PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

| | |
|---|---|
| In re application of | |
| Niels Gebauer | Examiner G. Robinson |
| Serial No. 09/189,615 | Group Art Unit 2168 |
| Filing Date: 11/09/98 | Docket No. 33012/246/101 |
| For: METHOD AND APPARATUS PROVIDING AN AVAIL-ABILITY MESSAGE TO REMOTE USER TERMINAL (Amended) | **FOURTH SUPPLEMENTAL APPEAL BRIEF** |

---

**APPELLANT'S THIRD SUPPLEMENTAL BRIEF**
**FILED UNDER 37 C.F.R. § 41.37**

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

**CERTIFICATE UNDER 37 C.F.R. 1.8:** I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to the: Commissioner for Patents, Alexandria, VA, 22313-1450 on this 10th day of December, 20 07.

By Carolyn I. Erickson

This fourth supplemental appeal brief is being filed in triplicate within thirty days of the Notification of Non-Compliant

Appeal mailed November 8, 2007. Permission is hereby granted to charge or credit deposit account number 14-0620 for any errors in fee calculation. Appellants request this Supplemental Appeal Brief be made of record and fully considered.

## REAL PARTY IN INTEREST

The Real Party in interest is:

Unisys Corporation

Township Line and Union Meeting Roads

Blue Bell, Pennsylvania 19424

being the assignee of the entire right, title, and interest by all inventors, by way of assignment documents filed at Reel 9576, frame 0750, in the United States Patent and Trademark Office.

## RELATED APPEALS AND INTERFERENCES

There are no known pending Appeals and/or Interferences which will directly affect or be directly affected by or have a bearing on the Board's decision in this appeal. Therefore, there are no decisions to be placed in the attached Related Proceedings Appendix.

**TABLE OF CONTENTS**

## STATUS OF CLAIMS

The subject patent application was filed on November 9, 1998 containing claims 1-20. Amendments to the claims and/or specification and drawings were filed on March 28, 2001, September 14, 2001, April 4, 2002, July 6, 2002, December 16, 2002, January 17, 2003, May 23, 2003, July 18, 2003, February 6, 2004 July 12, 2004, March 1, 2005, July 20, 2005, September 20, 2005, March 6, 2006, and July 19, 2006. Claims 21-22 were newly presented. It is believed that all claim amendments have been entered. The attached claims appendix contains finally rejected pending claims 1-22, being all pending claims, in their present form as finally rejected by final official action of the Examiner mailed May 31, 2006.

## STATUS OF THE AMENDMENTS

Applicants filed amendments on March 28, 2001, September 14, 2001, April 4, 2002, July 6, 2002, December 16, 2002, January 17, 2003, May 23, 2003, July 18, 2003, February 6, 2004 July 12, 2004, March 1, 2005, July 20, 2005, September 20, 2005, March 6, 2006, and July 19, 2006. These amendments were filed in response to the various official actions of the Examiner. The Examiner finally

rejected claims 1-22, being all pending claims by way of final office action mailed May 31, 2006. Applicants responded to this final office action by way of Amendment After Final under 37 C.F.R. 1.116 filed July 19, 2006. Though the Examiner has entered this amendment, claims 1-22 remain finally rejected.

## SUMMARY OF CLAIMED SUBJECT MATTER [1]

The present invention generally relates to data base management systems and more particularly relates to enhancements for providing access to data base management systems via internet user terminals[2]. The major advantage of the Internet is its universality. Nearly anyone, anywhere can become a user. That means that virtually all persons are potentially Internet users without the need for specialized training and/or proprietary hardware and software. One can readily see that providing access to a proprietary data base management system, such as Classic MAPPER, through the Internet would yield an extremely inexpensive and universally available means for accessing the data which it contains and such access would be without the need for considerable specialized training[3].

A special problem related to the Internet protocol is the inherent inability of the Internet to describe the status of a requested resource. In general, the only information that the Internet can determine from the unavailability of a requested

---

[1] The references to the specification and drawings provided herein are only exemplary and are not deemed to be limiting. The purpose of the references is to enable the Board to more quickly determine where the claimed subject matter is described within the present application.

[2]See Specification at page 3 lines 3-5.

[3]See Specification at page 4, line 19, through page 5, line 2.

resource is that the requested resource is unavailable. Ordinarily, this is accomplished by a simple time-out. However, if the requestor is provided no further information, it is unknown whether the resource is unavailable because it is busy, unavailable because it is not functioning, unavailable because it has been improperly addressed, unavailable because it does not exist, etc. As a result, the user is given no help with regard to what should be done concerning the unavailability of the requested resource[4].

For many Internet transactions, this lack of information is acceptable, because when making a resource request, the user is often searching for something without knowing exactly what type of response to expect. Unavailability simply means to continue the search elsewhere. However, users of existing proprietary data base management systems need to be provided with further information, because they are accustomed to utilizing a dedicated resource having defined and known characteristics. They are not at liberty to simply search elsewhere. They need to know that they have properly addressed an existing resource. If the resource is unavailable because it is busy or has failed, they need to know when to try again[5].

The present invention overcomes the disadvantages of the prior art by providing a method of and apparatus for utilizing the power

---

[4]See Specification at page 6, lines 14-22.
[5]See Specification at page 7, lines 1-9,

of a full featured data base management system by a user at a terminal coupled to the world wide web or internet. In order to permit any such access, the present invention must first provide a user interface, called a gateway, which translates transaction data transferred from the user over the internet in HTML format into a format from which data base management system commands and inputs may be generated[6]. To make access to a proprietary data base by Internet users practical, a sophisticated security system is also required to prevent intentional or inadvertent unauthorized accesses[7].

Given that the gateway, security system, and service processing combine to provide the internet terminal with the full features of the existing proprietary data base management system, the data base management system client can conveniently access the data base from either an existing dedicated terminal or from an internet terminal. The preferred mode of the present invention provides an additional feature to make the usage comparable from either terminal. A message may be stored within the repository to notify potential users of the availability status of the data base management system. This message may be composed and/or modified by

[6]See Specification at page 8, lines 3-9,
[7]See Specification at page 8, lines 15-17.

the administrator to provide the user with whatever information may be deemed appropriate in view of the unavailability[8].

The administrator creates a text file containing the desired message to be stored in the repository. An object is created whereby the service handler converts the text file to an HTML page and returns it to a requestor upon the occurrence of one or more predetermined conditions. Typical conditions include, system maintenance precluding availability, extensive user queuing, and major data base updating[9].

Claims 16 and 18 are the only pending claims introducing "means-plus-function" limitations. Claim 16 has four such limitations which are correlated to Applicants' disclosure as follows:

a) "permitting means for permitting a user to interact with a digital data base by generating a service request in anticipation of a response"[10];

b) "providing means responsively coupled to said permitting means for providing said user with access to a publicly accessible digital communication network via service-based requests"[11];

c) "offering means responsively coupled to said permitting means for offering data processing services according to dialog-

---

[8]See Specification at page 10, lines 10-18,
[9]See Specification at page 10, lines 19-23.
[10]See Specification at page 19, lines 4-6 and Fig. 4, element 54.
[11]See Specification at page 19, lines 4-6 and Fig. 4, element 66.

based requests if available by honoring said service request to generate said response"[12]; and

d) "notifying means responsively coupled to said offering means and said permitting means for notifying said user by transfer of an HTML display page in response to said service request of the unavailability of said offering means when said offering means is unavailable and that said service request will not be honored unless subsequently reinitiated at a time during which said offering means is available"[13].

Claim 18 has a single "means-plus-function" element: "storing means for storing a predefined unavailability message as a text file"[14].

In accordance with the Notification of Non-Compliant Appeal Brief mailed July 12, 2007, Applicants herewith endeavor to "separately map each independent claim to the specification by page and line number and to the drawings if any".

Claim 1:

--- element a -- see Figs. 4 and 11 and specification at pages 20-21.

Claim 6:

---

[12]See Specification at page 19, lines 2-4 and Fig. 4, element 62.
[13]See Specification at page 21, lines 16-23 and page 36, lines 2-7; and Figs.4 and 11.
[14]See Specification at page 20, lines 1-2, and Fig. 4, element 80.

--- element a -- see Fig. 4, element 54, and specification at page 19, lines 4-6;

--- element b -- see Fig. 4, elements 66, and specification at page 19, lines 4-6;

--- element c -- see Fig. 4, element 62, and specification at page 19, lines 2-4;

--- element d -- see Fig. 4, element 62, and specification at page 19, lines 2-4;.and.

--- element e -- see Figs. 4 and 1, and specification at page 21, lines 16-23 and page 36, lines 2-7.

Claim 11:

--- element a -- see Fig. 4, element 66, and specification at page 19, lines 4-6;

--- element b -- see Fig. 4, element 62, and specification at page 19, lines 2-4;

--- element c -- see Fig. 4, element 62, and specification at page 19, lines 2-4; and

--- element d -- see Figs. 4 and 11, and specification at page 21, lines 16-23, and page 36, lines 2-7.

Claim 21:

--- element a -- see Fig. 4, element 54, and specification at page 19, lines 4-6;

--- element b -- see Fig. 4, element 66, and specification at page 19, lines 4-6;

--- element c -- see Fig. 4, element 62, and specification at page 19, lines 2-4;

--- element d -- see Fig. 4, element 80, and specification at page 20, lines 1-2; and

--- element e -- see Figs. 4 and 11, and specification at page 21, lines 16-23, and page 36, lines 2-7.

## GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

1. Are claims 1-22 unpatentable under 35 U.S.C. 103(a) as being obvious over COOL ICE User's Guide release 1.0 (hereinafter referred to as "Unisys") in view of U.S. Patent No. 6,094,659, issued to Bhatia (hereinafter referred to as "Bhatia")?

## ARGUMENT

**I. Claims 1-22 are not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Claims 1-22 have been rejected under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia. This ground of rejection should be reversed for failure of the Examiner to make a *prima facie* case of obviousness as specified by MPEP 2143.

To make a *prima facie* case of obviousness, MPEP 2143 requires the Examiner to provide evidence and argument showing: 1)motivation to make the alleged combination; 2)reasonable likelihood of success of the alleged combination; and 3)all claimed elements within the alleged combination. The Examiner has failed to make any of these three required showings for any of the rejected claims. Therefore, because the Examiner has not made a *prima facie* case of obviousness, Applicants need not and indeed cannot offer appropriate evidence and argument in rebuttal.

The first showing required of MPEP 2143 is that of "motivation". In her only apparent attempt at showing motivation, the Examiner states:

> It would have been obvious to one of ordinary skill at the time of the invention to have combined Bhatia with Cool ICE User's Guide because **Bhatia is concerned with informing a user of a failure condition through a set of predefined messages and a status message** is an important message that can be implemented with a high level language for communication. (Emphasis added)

This statement is clearly erroneous. Bhatia is concerned with a "Web Server for use in a LAN Modem"[15]. Therefore, the only "failure condition or other operational event" of concern to Bhatia is directly associated with the LAN Modem. The Abstract provides in part:

> The resulting page informs a user stationed at the workstation of a **failure condition or other operational event that then occurred at the LAN modem**. (Emphasis added)

Unisys makes no mention of a LAN (LOCAL AREA NETWORK) or LAN modem. Furthermore, Unisys has no need for a LAN or a LAN modem. Therefore, Applicant strongly disagrees that anyone practicing Unisys would have any motivation to employ a LAN or LAN modem. Such an element would be clearly superfluous to the teaching of Unisys.

Applicant made this argument to the Examiner in the Amendment After Final filed July 19, 2005. By way of response, the Examiner acknowledges her confusion with regard to the LAN of Bhatia and the

[15]See Title.

claimed "publicly accessible digital data communication network". She states in her Advisory Action mailed August 4, 2006:

> ...i.e., public digital communication network suggests LAN structure....

Though it may be theoretically possible for a LAN to become a "public digital communication network" as suggested by the Examiner, those of skill in the art would readily appreciate that the reason for use of a LAN is to provide a **private** network.

Having failed to show any motivation for the alleged combination, the Examiner completely ignores her obligation to show reasonable likelihood of success. Most probably she has failed to do so because there is not reasonable likelihood of success.

MPEP 2143.03 requires that all claim limitations must be taught or suggested by the alleged combination. It reads in part:

> To establish *prima facie* obviousness of a claimed invention, all the claim limitations must be taught or suggested by the prior art. *In re Royka,* 490 F2d. 981, 180 USPQ 580 (CCPA 1974), "**All words in a claim must be considered in judging the patentability of that claim against the prior art**". *In re Wilson,* 424 F.2d 1382, 1385, 165 UPQ 494, 496 (CCPA 1970). (emphasis added)

The Examiner has failed to meet the requirement to show all claim limitations within the alleged combination, because she has at least not considered "all words in a claim" as specifically required by MPEP 2143.03.

**IA. Claim 1 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Claim 1, for example, is limited by transferring "**an unavailability message** as an HTML display page to said user terminal **in response to said service request** when **said data base management system** is unavailable to receive and respond to said service request". The Examiner admits that Cool ICE does not contain this limitation. Therefore, she clearly erroneously states:

Bhatia teaches this feature.

This statement is clearly erroneous in view of the disclosure of Bhatia which has no "unavailability message", no "response to said service request", no "data base management system", and no "unavailable to receive and respond to said service request". The Examiner was respectfully reminded that MPEP 2143.03 requires "**All words in a claim must be considered**". Nevertheless, she has continues to ignore much of the language and therefore the basis of claim 1.

The rejection of claim 1, and all claims depending therefrom, should be reversed for failure of the Examiner to meet any of the three required showings specified by MPEP 2143.

**IB. Claim 2 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Claim 2 depends from claim 1 and is further limited by "wherein said data base management system includes a repository for storing said unavailability message as a text file". In making her rejection, the Examiner states:

> Regarding claim 2: a repository for storing said unavailability message [note: Bhatia, Figure 20 Repository of Documents (1860); col. 60 lines 17-25]

The claim specifically requires "wherein said data base management system includes a repository". Bhatia contains no "data base management system" as claimed. Therefore, the Examiner simply ignores the requirement in contravention of MPEP 2143.03 which obligates the Examiner by stating that "**all words in a claim must be considered**". The rejection of claim 2 should be reversed for failure of the Examiner to apply MPEP 2143.03.

**IC. Claim 3 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Claim 3 depends from claim 2 and further limits the coupling network. As explained above, the alleged combination cannot meet the limitations of claim 2 from which claim 3 depends. Therefore, the alleged combination cannot meet the further limitations of claim 3. The rejection of claim 3 should be reversed.

**ID. Claim 4 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Claim 4 depends from claim 3 and is further limited by "wherein said repository includes space for storage of at least one variable for said unavailability message permitting an administrator to modify said unavailability message". Having expressly found that Cool ICE has no "unavailability message", the Examiner has somehow found that Cool ICE has the capability "permitting an administrator to modify said unavailability message". Not only is this finding clearly erroneous, it defies common logic. The rejection of claim 4 should be reversed.

**IE. Claim 5 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Claim 5 depends from claim 4 and further limits the claimed data base management system. As explained above, the alleged combination cannot meet the limitations of claim 4 from which claim 5 depends. Therefore, the alleged combination cannot meet the further limitations of claim 5. The rejection of claim 5 should be reversed.

**IF. Claim 6 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 6, for example, is limited by an "administration management system", which is not even acknowledged by the Examiner. The rejection of claim 7 should be reversed as being improperly examined.

**IG. Claim 7 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an

> unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 7, for example, is limited by an "wherein said data base management system has a repository having storage for a text file containing said unavailability message", which is not even acknowledged by the Examiner. The rejection of claim 7 should be reversed as being improperly examined.

**IH. Claim 8 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request

> [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 8, for example, is limited by an "wherein said repository has storage for a variable to be included in said unavailability message to permit an administrator to change said unavailability message", which is not even acknowledged by the Examiner. The rejection of claim 8 should be reversed as being improperly examined.

**II. Claim 9 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 9, for example, is limited by an "wherein said publicly accessible digital communications network is the world wide web", which is not even acknowledged by the Examiner. The rejection of claim 9 should be reversed as being improperly examined.

**IJ. Claim 10 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does

not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 10, for example, is limited by an "wherein said user terminal is an industry compatible personal computer having a commercially available web browser", which is not even acknowledged by the Examiner. The rejection of claim 10 should be reversed as being improperly examined.

**IK. Claim 11 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 11, for example, is limited

by an "determining via an administration management system whether said data base management system is currently capable of honoring said service request", which is not even acknowledged by the Examiner. The rejection of claim 11 should be reversed as being improperly examined.

**IL. Claim 12 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 12, for example, is limited by an "wherein said transferring step further comprises transferring said unavailability message to said user terminal via

said publicly accessible digital data network", which is not even acknowledged by the Examiner. The rejection of claim 12 should be reversed as being improperly examined.

**IM. Claim 13 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 13, for example, is limited by an "wherein said transferring step further comprises adding a variable to said unavailability message to permit an administrator to modify said unavailability message", which is not even

acknowledged by the Examiner. The rejection of claim 13 should be reversed as being improperly examined.

**IN. Claim 14 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 14, for example, is limited by an "wherein said publicly accessible digital data network further comprises the world wide web", which is not even acknowledged by the Examiner. The rejection of claim 14 should be reversed as being improperly examined.

**IO. Claim 15 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 15, for example, is limited by an "wherein said data base management system further comprises a commercial data base management system", which is not even acknowledged by the Examiner. The rejection of claim 15 should be reversed as being improperly examined.

**IP. Claim 16 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 16, for example, is limited by an "offering means responsively coupled to said permitting means for offering data processing services according to dialog-based requests if available by honoring said service request to generate said response", which is not even acknowledged by the Examiner. The rejection of claim 16 should be reversed as being improperly examined.

**IQ. Claim 17 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 17, for example, is limited by an "wherein said publically accessible digital communication network further comprises the world wide web", which is not even acknowledged by the Examiner. The rejection of claim 17 should be reversed as being improperly examined.

**IR. Claim 18 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 18, for example, is limited by an "wherein said offering means further comprises storing means for storing a predefined unavailability message as a text file", which is not even acknowledged by the Examiner. The rejection of claim 18 should be reversed as being improperly examined.

**IS. Claim 19 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 19, for example, is limited by an "wherein said offering means further comprises a commercial data base management system", which is not even acknowledged by the Examiner. The rejection of claim 19 should be reversed as being improperly examined.

**IT. Claim 20 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The

Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 20, for example, is limited by an "wherein said permitting means further comprises an industry standard personal computer", which is not even acknowledged by the Examiner. The rejection of claim 20 should be reversed as being improperly examined.

**IU. Claim 21 is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.**

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 21, for example, is limited by an "an HTML display page containing an unavailability message generated by said data base management system which notifies said

human user of unavailability of said data base management system and that said service request will not be honored unless reinitiated at a subsequent time when said data base management system is available", which is not even acknowledged by the Examiner. The rejection of claim 21 should be reversed as being improperly examined.

**IV. Claim 22** is not unpatentable under 35 U.S.C. 103(a) as being obvious over Unisys in view of Bhatia.

Instead of examining claims 6-22, which have differing statutory and judicial bases of patentability as well as differing claim limitations, the Examiner simply states:

> The limitations of claims 6-22 have been addressed above in claims 1-5, except for the following: transferring an unavailability message to said user terminal if said determining step determines data base management system is not currently capable of honoring said service request [note: Bhatia Figure 4B; col. 7 lines 4-26 protocol may be event-specific; col. 24 lines 23-39].

In addition to this statement being legally and grammatically incorrect, to the extent understandable, it is clearly erroneous. Furthermore, the statement is legally irrelevant, because it does not address the language of any claim or limitation thereof. The Examiner is prohibited by MPEP 2143.03 from disregarding Applicants' claimed invention. Claim 22, for example, is limited by an "a repository for storage of a text file containing said unavailability message", which is not even acknowledged by the

Examiner. The rejection of claim 22 should be reversed as being improperly examined.

## CONCLUSION

Having thus reviewed the final rejections of claims 1-22, being all pending claims, it seems abundantly clear that the limitations of these claims are not unpatentable in view of the prior art of record. Thus, the rejection of these claims should be reversed as being based upon clearly erroneous fact findings and errors of law.

Respectfully submitted

Niels Gebauer

By his attorney,

Date Dec 10, 2007

Wayne A. Sivertson
Reg. No. 25,645
Suite 401,Broadway Place East
3433 Broadway Street N.E.
Minneapolis, Minnesota 55413
(612) 331-1464

## CLAIMS APPENDIX

1. In a data processing environment having a user terminal operated by a user which generates a service request coupled to a publicly accessible digital communications network and having a data base management system which receives and responds to said service request when available by execution of an ordered sequence of command language script, the improvement comprising:

a server coupled to said user terminal via said publicly accessible digital communications network and coupled to said data base management system wherein said server includes an administration management system which transfers an unavailability message as an HTML display page to said user terminal in response to said service request when said data base management system is unavailable to receive and respond to said service request which signifies to said user that said service request will not be honored unless reinitiated at a subsequent time when said data base management system is available to honor said service request.

2. The improvement according to claim 1 wherein said data base management system includes a repository for storing said unavailability message as a text file.

3. The improvement according to claim 2 wherein said publicly accessible digital communications network is the world wide web.

4. The improvement according to claim 3 wherein said repository includes space for storage of at least one variable for said unavailability message permitting an administrator to modify said unavailability message.

5. The improvement according to claim 4 wherein said data base management system is a commercial data base management system.

6. An apparatus comprising:

a. a user terminal operated by a user which generates a service request;

b. a publicly accessible digital communications network coupled to said user terminal;

c. a server coupled to said user terminal via said publicly accessible digital communications network;

d. a data base management system coupled to said server which responds to said service request if available by execution of an ordered sequence of command language script; and

e. an administration management system coupled to said data base management system and said server which transfers an HTML display page containing an unavailability message from said server to said user terminal in response to said service request when said data base management system is not available to indicate unavailability of said data base management system which notifies said user that said service request will not be honored unless reinitiated at a subsequent time when said data base management system is available to honor said service request.

7. The apparatus of claim 6 wherein said data base management system has a repository having storage for a text file containing said unavailability message.

8. The apparatus of claim 7 wherein said repository has storage for a variable to be included in said unavailability message to permit an administrator to change said unavailability message.

9. The apparatus of claim 8 wherein said publicly accessible digital communications network is the world wide web.

10. The apparatus of claim 9 wherein said user terminal is an industry compatible personal computer having a commercially available web browser.

11. A method of communicating between a user terminal operated by a user and a data base management system comprising:

a. transmitting a service request from said user terminal via a publicly accessible digital data communication network to said data base management system;

b. determining via an administration management system whether said data base management system is currently capable of honoring said service request;

c. honoring said service request by the execution of an ordered sequence of command language statements by said data base management system if said determining step determines that said data base management system is currently capable of honoring said service request; and

d. transferring an HTML display page containing an unavailability message from said administration management system to said user terminal if said determining step determines that said data base management system is not currently capable of honoring said service request which notifies said user that said service request will not be honored unless subsequently

transmitted at a future time when said data base management system is available to honor said service request.

12. A method according to claim 11 wherein said transferring step further comprises transferring said unavailability message to said user terminal via said publicly accessible digital data network.

13. A method according to claim 12 wherein said transferring step further comprises adding a variable to said unavailability message to permit an administrator to modify said unavailability message.

14. A method according to claim 12 wherein said publicly accessible digital data network further comprises the world wide web.

15. A method according to claim 14 wherein said data base management system further comprises a commercial data base management system.

16. An apparatus comprising:

a. permitting means for permitting a user to interact with a digital data base by generating a service request in anticipation of a response;

b. providing means responsively coupled to said permitting means for providing said user with access to a publicly accessible digital communication network via service-based requests;

c. offering means responsively coupled to said permitting means for offering data processing services according to dialog-based requests if available by honoring said service request to generate said response; and

d. notifying means responsively coupled to said offering means and said permitting means for notifying said user by transfer of an HTML display page in response to said service request of the unavailability of said offering means when said offering means is unavailable and that said service request will not be honored unless subsequently reinitiated at a time during which said offering means is available.

17. An apparatus according to claim 16 wherein said publically accessible digital communication network further comprises the world wide web.

18. An apparatus according to claim 17 wherein said offering means further comprises storing means for storing a predefined unavailability message as a text file.

19. An apparatus according to claim 18 wherein said offering means further comprises a commercial data base management system.

20. An apparatus according to claim 19 wherein said permitting means further comprises an industry standard personal computer.

21. An apparatus comprising:

a. a user terminal providing access by a human user for generating a service request;

b. a publicly accessible digital communications network responsively coupled to said user terminal;

c. a server responsively coupled to said user terminal via said publicly accessible digital communications network;

d. a data base management system responsively coupled to said server which honors said service request when available; and

e. an HTML display page containing an unavailability message generated by said data base management system which notifies said human user of unavailability of said data base management system and that said service request will not be honored unless reinitiated at a subsequent time when said data base management system is available.

22. An apparatus according to claim 21 further comprising a repository for storage of a text file containing said unavailability message.

## EVIDENCE APPENDIX

During the prosecution of the subject application, the following three (3) declarations were submitted resulting in the removal of U.S. Patent No. 6,347,330, issued to Dawson et al. as a reference applicable against the subject application:

1. Declaration under 37 C.F.R. 1.132 of Barbara A. Christensen, dated August 6, 2002 and filed August 6, 2002;

2. Declaration under 37 C.F.R. 1.131 of Niels Gebauer, dated December 11, 2002 and filed December 16, 2002; and

3. Declaration under 37 C.F.R. 1.131 of Niels Gebauer, dated January 13, 2003 and filed January 17, 2003.

There is no other evidence or documents deemed appropriate to be included within this Appendix.

## RELATED PROCEEDINGS APPENDIX

There are no decisions or other papers deemed appropriate to be included in this Appendix.

PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of )
)
Niels Gebauer ) Examiner G. Robinson
Serial No. 09/189,615 ) Group Art Unit 2177
)
Filing Date: 11/09/98 )
) Docket No.
) 3012/246/101
)
For: METHOD AND APPARATUS FOR) **DECLARATION**
PROVIDING AN AVAIL- ) **UNDER**
ABILITY MESSAGE TO RE- ) **37 C.F.R. 1.132**
MOTE USER TERMINAL )
(Amended) )

Honorable Commissioner of Patents
and Trademarks
Washington, D.C. 20231

Dear Sir:

**CERTIFICATE UNDER 37 C.F.R. 1.8:** I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to the: Assistant Commissioner for Patents, Washington, D.C., 20231 on this 16th day of December 20 02.

By John L. Rooney
John L. Rooney

**DECLARATION**

The undersigned declares as follows:

1. My name is Niels Gebauer;

2. My home address is 8/86 Milson Road, Cremorne Point, NSW 2090, Australia;

3. I am employed by Unisys Corporation, assignee of the subject invention, as Software Engineer ;

4. I am the sole inventor of the U.S. Patent Application Serial No. 09/189,615 filed November 11, 1998;

5. The invention of claims 1-22 was first embodied in a commercial product of Unisys Corporation entitled Cool ICE Revision 1.1 and was therefore completely reduced to practice with that product;

6. Cool ICE Revision 1.1 was first placed on commercial sale on November 14, 1997;

7. Further declarant sayeth not.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon, I further declare that I understand the content of this declaration.

Date 11/12-02

Niels Gebauer

N. Gebauer

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In application of )
)
Niels Gebauer )
)
Serial No. 09/189,615 )
)
Filing Date: 11/09/98 )
)
For: METHOD AND APPARATUS FOR PROVIDING AN AVAILABILITY MESSAGE TO REMOTE USER TERMINAL (Amended) )

Examiner G. Robinson

Group Art Unit 2177

Docket No. 33012/246/101

**DECLARATION UNDER 37 C.F.R. 1.131**

Honorable Commissioner of Patents
and Trademarks
Washington, D.C. 20231

Dear Sir:

**CERTIFICATE UNDER 37 C.F.R. 1.10**: The undersigned hereby certifies that this paper is being deposited in the United States Postal Service, "Express Mail Post Office to Addressee" having an Express Mail mailing label number of: EL 101 812 958 US, in an envelope addressed to Assistant Commissioner for Patents, Washington, D.C. 20231, on this 17th day of January, 2008

By _____
Carolyn I. Erickson

**DECLARATION**

The undersigned declares as follows:

1. My name is Niels Gebauer;

2. My home address is 8/86 Milson Road
Cremorne Point
NSW 2090
Australia;

1. I am employed by Unisys Corporation, assignee of the subject invention, as Software Engineer;

4. I am the sole inventor of the U.S. Patent Application Serial No. 09/189,615 filed November 11, 1998;

5. The invention of pending claims 1-22 of the subject U.S. Patent Application was first commercially embodied in a product of Unisys Corporation entitled Cool ICE Revision 1.1;

6. Cool ICE Revision 1.1 was first placed on commercial sale on November 14, 1997;

7. The previously submitted declaration under 37 C.F.R. 1.132 of Barbara A. Christensen, along with its accompanying Exhibits, establishes the date of release for commercial sale of Cool ICE Revision 1.1;

8. Further declarant sayeth not.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon, I further declare that I understand the content of this declaration.

Date 13/Jan - 2003

N. Gebauer

Niels Gebauer

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of )
)
Niels Gebauer )
)
Serial No. 09/189,615 )
)
Filing Date: 11/09/98 )
)
For: METHOD AND APPARATUS FOR PROVIDING AN AVAIL-ABILITY MESSAGE TO REMOTE USER TERMINAL (Amended) )

Examiner G. Robinson

Group Art Unit 2177

Docket No. 33012/246/101

**DECLARATION UNDER 37 C.F.R. 1.131**

Honorable Commissioner of Patents
and Trademarks
Washington, D.C. 20231

Dear Sir:

**CERTIFICATE UNDER 37 C.F.R. 1.10:** The undersigned hereby certifies that this paper is being deposited in the United States Postal Service "Express Mail Post Office to Addressee" having an Express Mail mailing label number of: EL 101 812 958 US, in an envelope addressed to Assistant Commissioner for Patents, Washington, D.C. 20231, on this 7th day of January, 2008.

By _____
Carolyn I. Erickson

**DECLARATION**

The undersigned declares as follows:

1. My name is Niels Gebauer;

2. My home address is 8/86 Milson Road
Cremorne Point
NSW 2090
Australia;

1. I am employed by Unisys Corporation, assignee of the subject invention, as Software Engineer;

4. I am the sole inventor of the U.S. Patent Application Serial No. 09/189,615 filed November 11, 1998;

5. The invention of pending claims 1-22 of the subject U.S. Patent Application was first commercially embodied in a product of Unisys Corporation entitled Cool ICE Revision 1.1;

6. Cool ICE Revision 1.1 was first placed on commercial sale on November 14, 1997;

7. The previously submitted declaration under 37 C.F.R. 1.132 of Barbara A. Christensen, along with its accompanying Exhibits, establishes the date of release for commercial sale of Cool ICE Revision 1.1;

8. Further declarant sayeth not.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon, I further declare that I understand the content of this declaration.

Date 13/Jan - 2003

N. Gebauer

Niels Gebauer

PATENT

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re application of

Niels Gebauer

Serial No. 09/189,615

Filing Date: 11/09/98

For: METHOD AND APPARATUS FOR PROVIDING AN AVAILABILITY MESSAGE TO REMOTE USER TERMINAL (Amended)

Examiner G. Robinson

Group Art Unit 2177

Docket No. 33012/246/101

**DECLARATION UNDER 37 C.F.R. 1.132**

Honorable Commissioner of Patents and Trademarks
Washington, D.C. 20231

Dear Sir:

CERTIFICATE UNDER 37 C.F.R. 1.8: I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to the: Assistant Commissioner for Patents, Washington, D.C., 20231 on this 6th day of August, 2002.

By Melissa A. Abeldgaard
Melissa A. Abeldgaard

**DECLARATION**

The undersigned declares as follows:

1. My name is Barbara A. Christensen;

2. My home address is 6520 White Oak Rd.
Lino Lakes, MN 55038;

3. I am employed by Unisys Corporation, assignee of the subject invention, as Senior Software Engineer;

4. I have worked professionally with the Unisys Corporation family of products called Cool ICE and its predecessor family of products called MAPPER since 1978;

5. I have read claims 1-22 which are pending with regard to the subject U.S. Patent Application;

6. The invention of pending claims 1-22 of the subject U.S. Patent Application was first commercially embodied in Cool ICE Revision 1.1;

7. I am very familiar with Cool ICE Revision 1.1, a commercial product of Unisys Corporation, and its associated documentation including the "Cool ICE 1.1 Feature Specification", having a Release Date of October 31, 1997, attached hereto as Exhibit B;

8. The presence of the invention of pending claims 1-22 within Cool ICE 1.1 is established at pages 9 of 20, 15 of 20, and HDK 00077 of Exhibit B;

9. Cool ICE Revision 1.1 was first released as a commercial product on November 14, 1997;

10. Attached hereto as Exhibit A is a copy of the Meeting Minutes for November 6/7, 1997 of the Cool ICE and UnixWare group which established the General Commercial Availability date (i.e., GCA) of Cool ICE 1.1 as Friday, November 14, 1997; and

11. Further declarant sayeth not.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon, I further declare that I understand the content of this declaration.

Date 8/6/2002

Barbara A. Christensen

**Subject:**
**Cool ICE 1.1 for Windows NT and UnixWare Meeting minutes for November 6/7, 1997**

Hello everyone,

Discussion topics:

The final copy of Cool ICE 1.1 was shipped to POSM on Friday Oct. 31. The CD-ROM media has been returned and is being verified. Approval for manufacturing and shipment is expected to occur on Friday Nov. 7th.
The final version of the NT Cool ICE 1.1 is available on the FTP server.
The final two action items have been completed. Julian has provided FAQ information to Tom Johnson which will be put on the Cool ICE web site.
This is the FINAL Cool ICE 1.1 meeting. Any new problems or issues will be handled as support issues for the 1.1 product.

| Activities | Cool ICE 1.1 |
|---|---|
| Final Media ->POSM | Fri. 10/31 |
| Receive verify media | Thur. 11/06 |
| Approve media | Fri. 11/07 |
| **GCA** | **Fri 11/14** |

**There are no more meetings planned for Cool ICE 1.1.** Thanks for your cooperation and support for this product.

**Product: Cool ICE for Windows NT and UnixWare**
**Level: 1.1**
**Dependencies: NT MAPPER 5.3.3, UNIX MAPPER 5R3C, WebTx 3.0, CPIX 3.0 & CPNX 3.0**
**RTM (Release To Manufacturing): ~~9/29/97~~ 10/24/97**

- **Activity ID Activity DescriptionPlan DateReplan DateActual DateResponsible Organization**1ACUS Product Information Plan finalized970630970718970718PI Team2RSVL. Product Information Plan finalized970630970814 970828 PI Team3Product Test Plan finalized970630970718970722Test Team6Last FIX media delivery from Australia to Rsvl. Information Plan finalized970808597101797102 8Niels/Julian7Final Test of media in Rsvl problems to Australia Information Plan finalized970822971015*971030Test Team180Doc CRC to RS970819970922PI Team190Engineering Ships CD-ROM Media to POSM970819971024971031Engineering300POSM Returns CD-ROM Media to Engineering970828971029971104POSM270Checkdisks and CD label verified970828971031971106Engineering2810Authorize Imation/POSM to manufacture CDs970829971031971107Engineering390Ship Authorization Received at POSM970909971031971107Rel Mgmt400Release to Manufacturing (RTM / GCA)970909971114971114POSMDependent on final media release on NT 5.3.3

**Action items** are summarized below.
**NumberResponsibilityAction ItemStatusComments** 970429-5Julian Watts**All Requirements:** All Engineering tasks, work efforts and individual assignments to be sized and prepared by Julian Watts for the team to review and utilize as the basis for the Cool ICE 1.1 PDP. 970429**Closed.** May 22. 970429-6Julian Watts**All Requirements:** Functional and Design Specifications will be need to be prepared for the project. Julian Watts to check with ACUS PI for assistance in this area.970429**Closed.** June 05.

# Feature Specification
# Cool ICE 1.1

**Document Number:** FS-ICE11

Revision Level: -

Release Date: 31-Oct-1997

Document Author: Niels Gebauer

Project: Cool ICE 1.1

Document Path: C:\documents\CoolICE

Document Filename: FS-ICE11.doc

| Section | Status |
|---|---|
| 1. Functional Section | Please Review ☐ Inspected ☐ Under rework ☐ |
| 2 Detailed Design | Please Review ☐ Inspected ☐ Under rework ☐ |

# 1. Functional Section

Cool ICE 1.1 is the second release of Cool ICE and is a feature rich release.

## 1.1 High Level Functional Description

New features have been required in the following 3 main areas of Cool ICE:

- **Organize & Manage**
  - Maintaining Settings at System level
  - Improved GUI Interface
  - Service Templates
  - Importing Existing Runs
  - Expanded Repository Capability for storing Objects
  - Security Enhancements
  - Managing Static Documents
  - Arranging list of Categories and Services
  - Customizing Icons for Categories and Services
  - File Transfer between Workstation and Server
  - Drawer Browsing Capability
  - Service Expiration
- **Service Development**
  - Additional System Variables
  - Enhanced Integration to HTML Authoring Tools
  - Image Maps Capability
  - Creating Services based on Templates
  - Cookies Support
  - Uploading Files
  - Downloading Files
- **Deployment**
  - Enhance Session Management
  - Controlling System Availability & System Trace
  - Enhanced Graphics Support
  - Positional URL Parameters
  - Service Error Analyzer via the Browser
  - Additional System Variables
  - Support for Image Maps
  - Service Expiration Date Enforcement
  - Automatic Object Download
  - Image Viewer
  - User Password Change
  - Cookies Support
  - Repository Search Capability
  - Event Viewer
  - Uploading Files to the Repository
  - Enhanced Handling of HTML Tag Delimiters

## 1.2 Requirements and Reference Documents

The following table lists the requirements for Cool ICE 1.1 in the order they have been received. This list is a subset of the total requirements for Cool ICE and it only includes requirements specifically for the Cool

ICE Runware component.

**Cool ICE 1.1- Requirements**

| Req. No. | Description |
|---|---|
| 1 | **New Category attributes**. When setting up a category it will be possible to specify the purpose/content of the category such as:-<br>- **Template category** (When building services, this would allow to select a template that come close to what the user want to do (i.e. different forms and different kinds of database access)<br>- **Image category** (This allow to store and reference Images in Cool ICE without the service developer has to develop special services top handle this)<br>- **Applet category** (This allow to store and reference Applets in Cool ICE without the service developers has to anything)<br>- **Normal Service category**<br>- **Cool ICE System category** (This allow the customers easily to take a copy of the Cool ICE system services and customize these services)<br>Other category attributes such Read Only category and Access (Direct/Indirect) will be possible. |
| 2 | Allow the users to **change** their **password**. This include specifying user-id's for which change of password is not allowed (such as Guest user-id).<br>A change of password needs to be communicated back to the Gateway so the gateway can update its session information. |
| 4 | **Import** HTML Template, optionally **modify Hyper Link** References and **Form Action Field** according to the ICE standard (<GURL>/Category/Service) |
| 12 | **Import existing MAPPER Runs** to the Repository. |
| 13 | **Enhance the look and feel of dialog boxes to MS Sans Serif non-bold.** |
| 14 | **Export and Import** of HTML to and from a service should be enhanced to allow for services that include **more than one HTML template!** |
| 15 | A function in ICE Administration to **define structure of Categories and Services.** In the current functionality a new Category and new Service is inserted at the bottom of the list. It will be required to specify some sort of structure in the list. |
| 18 | Add a new **Attribute for Categories and Services for specifying icon** to be displayed when building and displaying menu of categories and services. This is to provide an easy way of tailoring a look and feel for a user or group of users. |
| 19 | Enhance service development with features for maintaining a Library of reusable objects. This could be achieved through the use of template category as in Req. no 1. |
| 21 | **Search facility** to allow users to search for services in the repository. |
| 22 | **Cool ICE Admin interface available from a browser.**<br>- Initially only for selected administrative functions.<br>- View Event Log should be the first to implement. |
| 23 | **Enhance Export** service **function** and Source function enhanced to handle @brk .....@brk better. When in "Script on Mode", lines without @ must have an html tag delimiter (<) in order to change mode to "Script Off". |
| 25 | Function to take the whole **Cool ICE system On and Off the net.** |
| 26 | **Improve handling of positional parameters on the URL.** |
| 27 | When **saving a service**, verify that the original service report has not been change manually in the meantime by some one else. |
| 28 | **Web-TX shared functionality:**<br>**a) File upload to Cool-ICE**<br>**b) File download to browser from Cool-ICE**<br>**c) Session management** |
| 32 | Possibility to turn **Trace On/Off of the Cool ICE Service Handler.** |
| 34 | **Improve validation** of existence **of directories and files** on the NT and UNIX system. |
| 40 | **Single source for NT and UNIX** |

| 60 | Service Expiration Date enforcement |
|---|---|

### 1.2.1 Other Requirements that affect this implementation

Migration from Cool ICE 1.0 must be seamless and must be handled automatic by the setup procedure.

### 1.2.2 List of requirements which will not be met

All requirements for Cool ICE 1.1 have been met.

## 1.3 Dependencies

Cool ICE 1.1 is dependent of the following 3 main features being delivered by WebTx 3.0 and the associated MAPPER Gateway:

**1) Uploading files via the browser from the client workstation to Cool ICE.**
**2) Downloading files/objects from the Cool ICE repository to the client workstation specifying appropriate MIME type.**
**3) Secure Session Management.**

## 1.4 Definitions

| Term | Definition |
|---|---|
| Repository | Storage for Cool ICE Services and objects. |
| Service | Item in the repository containing server side script that executes when requested from a client browser. |
| Object | Common term for all items stored in the repository. An object can contain binary information as well textual information such as GIF/JPG images, Applets, Word document, Cool ICE Services and Static HTML Documents. |
| | |
| | |
| | |

## 1.5 Description of Users

There are three different types of users of Cool ICE:

1) Developers who develop and maintain Cool ICE Web based services.
2) People who will be administrating a Cool ICE Web Site.
3) End users who will be accessing a Cool ICE Web Site using a browser.

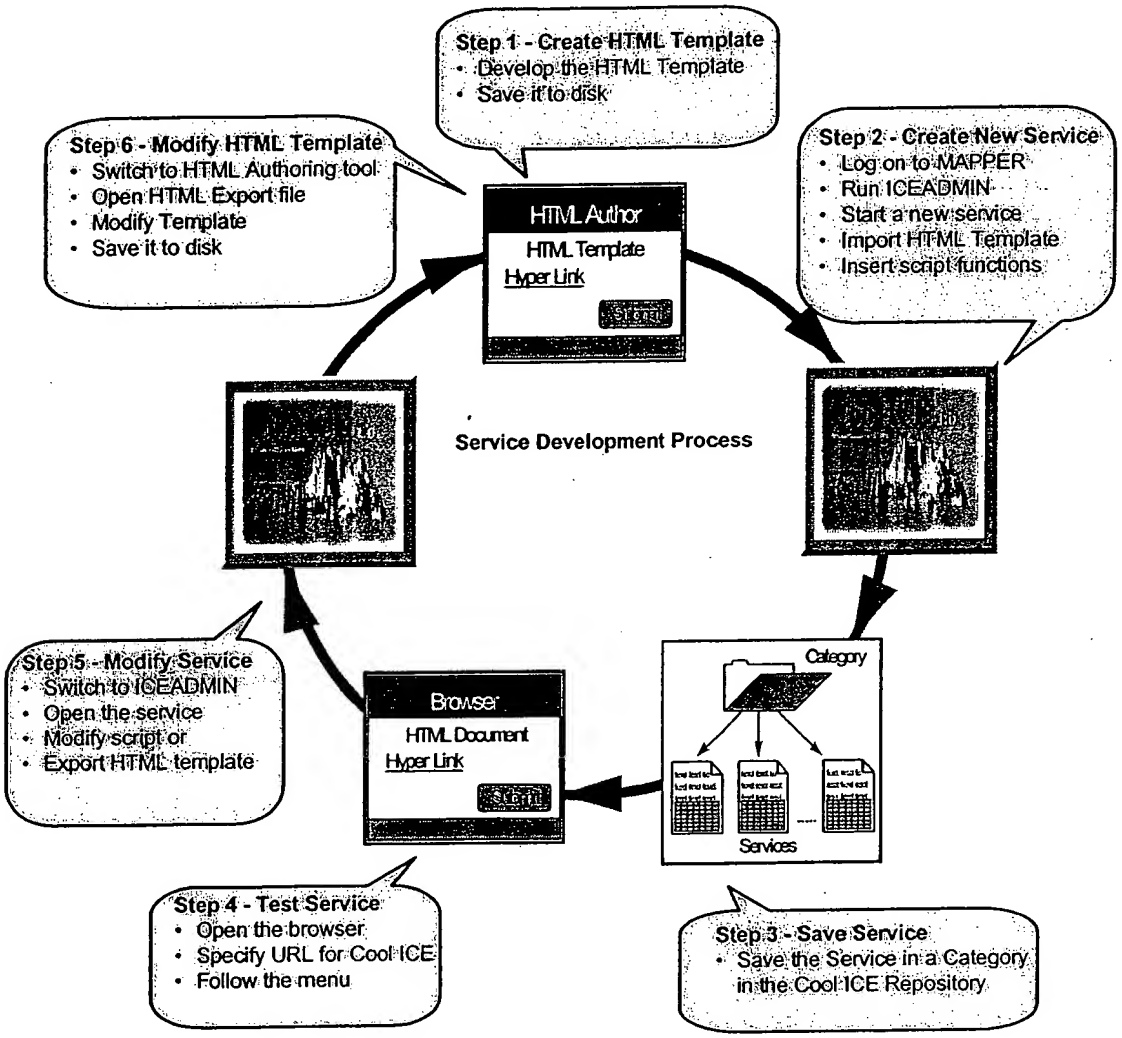
The same person will probably often perform the roles of category 1 and 2.

## 1.6 Areas yet to be addressed.

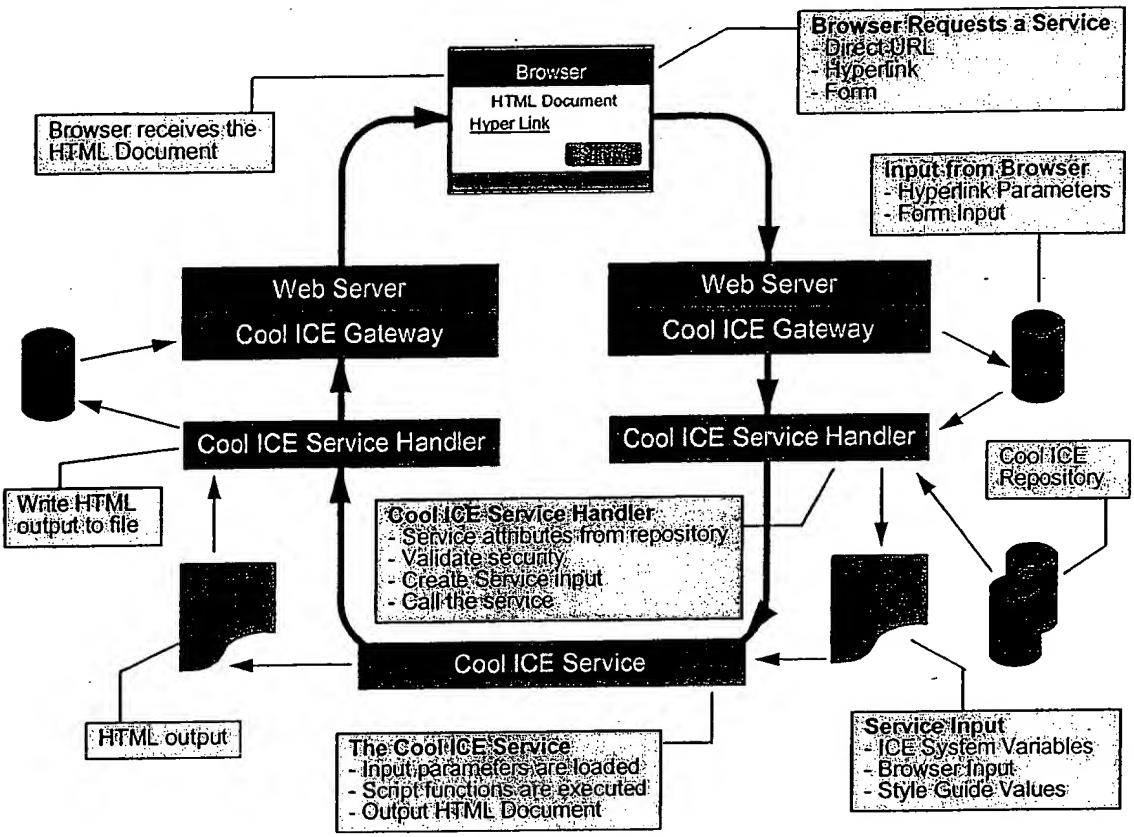
None.

## 1.7 Functional Flow

The following illustrate the main steps a developer would go through to develop a Cool ICE Web Service:



The following illustrate the flow of events from when a browser at a client workstation requests a Cool ICE Web service and until a response is received in the browser:




## 1.8 User Interface Description

Cool ICE has two different user interfaces:

1. Cool ICE Administration uses a traditional Microsoft Windows Graphical User Interface.
2. Cool ICE Web Services uses a Browser HTML Interface.

## 1.9 Resource estimate

Project Manager is responsible for ensuring this section is completed before sign-off.

| Task | Estimated Effort | Actual Effort |
|---|---|---|
| 1. Write Functional Section | | Should be actuals now |
| 2. Review Functional Section | | Should be actuals |
| 3. Write Functional Description Section | | |
| 4. Review Functional Description Section | | |
| 5. W rite Design Details Section | | |
| 6. Review Design Details Section | | |
| 7. Write Test Specification | | |
| 8. Review Test Section | | |
| 9. Coding Task | | |
| 10. Review Coding Task | | |
| **Total Person Costs** | Cost to Go = Total Hrs * $/person/hr | Cost so Far = Total Hrs * $/person/hr |

# 2. Detailed Design

## 2.1 Design Overview

The requirements for Cool ICE 1.1 have been grouped into the following three main areas:

- Organize & Manage
- Service Development
- Deployment

## 2.2 Organize & Manage

### 2.2.1 Maintaining Settings at System Level

This describes the implementation of requirements #25 and #32 specified in section 1.2 plus additional functionality derived from the requirements.

New Cool ICE Admin function "System Settings" will be added to the Options Menu to provide easy maintenance of system settings.

Table for Service handler settings will be added. Report 27E contains settings for configuring the Cool ICE Service Handler:-

- Trace - Trace On/Off
- Availability - On-Line/Off-Line

When taking the Cool ICE system off the net, a message can be entered which will be displayed to the browser users at deployment time.

In addition, the "System Settings" function will also allow specifying user-defined text to be displayed as the footer of all HTML pages. The Cool ICE Page footer, which is always put on all HTML pages, can now be turned On/Off and the text can be customized

### 2.2.2 Improved GUI Interface

This describes the implementation of requirement #13 specified in section 1.2 plus additional functionality derived from the requirements.

The Look and Feel of the Cool ICE Administration User Interface will be enhanced to use MS Sans Serif Non-Bold.

Function for creating a New Category will be enhanced to give a message after having completed creating the new category.

Function for creating a New Category will be enhanced to make the new category current if no other category is already open.

Function for Deleting a Category will be enhanced to give a message after having completed deleting the category.

Function for Removing a security profile will be enhanced to validate if the profile is allocated to any Categories or Services before allowing it to be removed.

The Category Export and Import functions will be enhanced to show a progress bar.

The Category Export function will be enhanced to verify if the path to export to already contain a previously exported category. The user will receive a warning and the option to overwrite.

The Category Import function will be enhanced to verify if the category to import to already contain objects. The user will receive a warning and the option to overwrite. This makes it easier to re-import a category without first removing the content.

Function for modifying security profile will be enhanced to modify the profile name every where it is used when the name of the profile is changed.

### 2.2.3 Service Templates

This describes the implementation of requirements #1 and #19 specified in section 1.2 plus additional functionality derived from the requirements.

Functions for creating and managing Cool ICE Services Templates will be added to Cool ICE Administration. This allows creating templates and subsequently allowing the service developer to create Cool ICE services based on available templates.

Templates provided a mechanism to share and reuse commonly used functions such as database access and HTML layouts.

Function "Template" for setting up templates will be added under menu item "New Repository Object".

### 2.2.4 Importing Existing Runs

This describes the implementation of requirement #12 specified in section 1.2 plus additional functionality derived from the requirements.

Function for Importing an existing MAPPER run and storing it in the repository as a Cool ICE Dynamic Service will be added to the Cool ICE Administration. This will allow browsing the MAPPER database for selection of the MAPPER Run to import. Option will be provided for selection of inserting standard Cool ICE Service header.

The function will be added as an additional option under menu item "New Repository Object", "Dynamic Service". The option will be called "Dynamic Service based on Existing Script".

### 2.2.5 Expanded Repository Capability for Storing Objects

This describes the implementation of requirements #1 and #19 specified in section 1.2 plus additional functionality derived from the requirements.

Functions for inserting Images and Applets and storing these as objects in the Cool ICE repository will be added to the Cool ICE Administration. This includes exporting/importing images and applets to/from the workstation. When an image or an applet stored in the Cool ICE repository is referenced from an HTML document, the object will automatically be down loaded to the browser. The Cool ICE Service Handler handles this.

Two new functions, "Image" and "Applet", will be added under menu item "New Repository Object".

In addition, a new Cool ICE System Service "DspImg" will be included to view an image stored in the Cool ICE Repository. The Cool ICE Menu Builder will be enhanced to automatically build a link to the new "DspImg" service when a GIF or JPG type object is direct accessible. This provides the capability to view images in the Cool ICE Repository just by changing the accessibility attribute to "Direct". Normally an image object will be accessible indirectly.

Functions for inserting Objects of any kind and storing these as objects in the Cool ICE repository will be added to Cool ICE Administration. This includes exporting/importing objects to/from the workstation. When an object stored in the Cool ICE repository is referenced, the object will automatically be downloaded to the browser. The Cool ICE Service Handler handles this. A function "Other Object" will be added under menu item "New Repository Object".

In addition, new function for maintaining a table of valid Cool ICE objects will be included. This allow to define the type of object and associated MIMI Type which is used at runtime when an object is downloaded via the browser to the end user workstation. A function "Object Types" will be added under menu item "Options".

### 2.2.6 Security Enhancements

This describes the implementation of requirements #1 and #2 specified in section 1.2 plus additional functionality derived from the requirements.

In addition to being able to store objects of any kind in the Cool ICE repository, the existing security profile mechanism will be enhanced to allow securing any object in the repository.

User Registration will be enhanced to allow specifying whether a user can change password or not. Users who are allowed to change their password can do so via the browser.

A "Change Password" service will be accessible from the browser for those users who have been granted the right to do so.

A new Cool ICE System Service "Change-Password-Form" will be included to allow Cool ICE browser users to change their own password. A user must be a registered Cool ICE user and be allowed to change password in order to use the "Change-Password" service. After successfully having changed the password, the user will be required to Signon again to Cool ICE. The Cool ICE signon form will automatically be displayed. This is necessary in order to signal to the MAPPER Gateway that the password has changed.

A new Cool ICE System Service "SignOff" will be included to allow Cool ICE users to sign off from Cool ICE. After Sign Off, the Cool ICE Sign-On form will be displayed. This means a user can safely leave the workstation without the workstation having a session open to Cool ICE.

### 2.2.7 Managing Static Documents

This describes the implementation of requirement #1 specified in section 1.2 plus additional functionality derived from the requirements.

Function for inserting Static HTML Documents and storing these, as objects in the Cool ICE repository will be added to Cool ICE Administration. This includes exporting/importing documents to/from the workstation. When a static document stored in the Cool ICE repository is referenced, the document will automatically be down loaded to the browser. The Cool ICE Service Handler handles this.

Function "Static Document" will be added under menu item "New Repository Object".

Importing HTML Documents has been enhanced to allow importing images being referenced by the Document. An option will be added to select that images being imported should be saved in the Cool ICE Repository. The original functionality of importing images to the Server Image Alias Directory will be maintained.

### 2.2.8 Arranging List of Categories and Services

This describes the implementation of requirement #15 specified in section 1.2 plus additional functionality derived from the requirements.

Functions for Arranging the list of Categories and Objects will be added to Cool ICE Administration. This provides the capability to move categories and objects around in order to specify the sequence in which they will be listed. The Menu builder uses the list of categories and services.

Function "Arrange Categories/Objects" will be added under menu item "Options".

### 2.2.9 Customizing Icons for Categories and Services

This describes the implementation of requirement #18 specified in section 1.2 plus additional functionality derived from the requirements.

New attributes for specifying customized icons for Categories and services will be added to System Settings table (27E). The menu builder will use these icons when displaying menu of categories and services. Default system icons will be used when nothing has been specified.

### 2.2.10 File Transfer between Workstation and Server

This describes the implementation of additional functionality derived from the requirements.

Function for transferring files between the workstation and the server will be included in Cool ICE Administration. Function "File Transfer" will be added under menu item "Options".

The primary purpose of the function is to provide an easy way of transferring images between the service developers workstation and the web server. The function, however, will allow transfer of any files. Browse capability will be provided for selecting files in any directory on the workstation. On the server, browsing will be restricted to directories listed in the Image Directory Alias table maintained through the Image Directory Alias function.

### 2.2.11 Drawer Browsing Capability

This describes the implementation of additional functionality derived from the requirements.

Creating a new category will be enhanced to include a Browse button for selecting and allocating a Drawer from the Cool ICE database.

The browse function will display a list of drawers in the database and allow selecting an open drawer. This makes it easy for users of Cool ICE who are not yet familiar with the Cabinet/Drawer structure of the database to setup a Cool ICE category.

The browse function is only available for browsing a database on the local Cool ICE system. The browse

button will be disabled when selecting to create a new category on a remote Cool ICE server.

### 2.2.12 Service Expiration

This describes the implementation of additional functionality derived from the requirements.

Function for listing services/objects that have expired or will expire within a specified number of days will be added.

Function "Service/Object Expiration" will be added under menu item "Options".

## 2.3 Service Development

### 2.3.1 Additional System variables

This describes the implementation of additional functionality derived from the requirements.

Additional Run-Time information about the environment will be made available to Cool ICE Services. The following Cool ICE system variables will be added to the System Variables section of the services input (-8) being passed to a service:-

- <GSysCat> Name of Cool ICE System category (f.ex.ICEADM)
- <GSvrName> Name of Web Server (f.ex. coolice.au.unisys.com)
- <GSvrProtocol> Protocol (f.ex. HTTP/1.0)
- <GSvrPort> Server Port number (f.ex. 80)
- <GRemoteAddr> IP Address of client (f.ex. 129.223.41.102)
- <GUserAgent> Client Browser (f.ex. Mozilla/3.01 (WinNT: I)
- <GGatewayIn> Gateway Input File
- <GInvkSignon> Invoke Signon (y/n)

### 2.3.2 Enhanced Integration to HTML Authoring Tools

This describes the implementation of requirements #4, #14 and #23 specified in section 1.2 plus additional functionality derived from the requirements.

Importing HTML Forms will be enhanced to modify the Form Action attribute to the standard Cool ICE way of referencing another service. "<GURL>/<category>/COOL-ICE-SERVICE" will be inserted in the action field. This means the service developer just has to modify "COOL-ICE-SERVICE" and insert the appropriate service name.

Exporting and Importing HTML Templates will be enhanced to allow for services that contain more than one HTML Template. If a service contain more than one HTML Template a dialog box will appear allowing the user to select which Template to export or import.

Importing HTML Templates or HTML Documents will be enhanced to allow importing images being referenced by the Template/Document. Optionally images can be imported and saved in the Cool ICE Repository. The original functionality of importing of images to the Server Image Alias Directory has been maintained.

The following additional enhancements will be implemented:

- Under special circumstances importing an HTML template containing Script causes the HTML delimiters <> not to be translated.
- Under special circumstances importing an HTML template containing very long lines causes truncation of the HTML input.

Exporting HTML will be improved to make a better decision whether a line is a Cool ICE Script or an HTML Tag particular when there is no @ character in column one.

### 2.3.3 Image Maps Capability

This describes the implementation of additional functionality derived from the requirements.

Support will be added for the HTML ISMAP's. The Cool ICE service will receive two variables (<X> and <Y>) containing the coordinate position of the mouse pointer when the user click within an image defined with the ISMAP attribute.

The Cool ICE Service Handler will be enhanced to create two variables (<X> and <Y>) in the service input report containing the corresponding x and y coordinate position of where the user clicked on the map.

### 2.3.4 Creating Services based on Templates

This describes the implementation of requirements #1 and #19 specified in section 1.2 plus additional functionality derived from the requirements.

Function for Creating a New Service Based on Cool ICE Template will be enhanced to allow the service developer to browse the repository to find and select a suitable template.

### 2.3.5 Cookies Support

This describes the implementation of requirement #28 specified in section 1.2 plus additional functionality derived from the requirements.

Support for cookies will be provided. Cookies can be set by individual Cool ICE services and the cookie keywords and associated values will be passed into the Cool ICE services. Cookies will be passed into the Cool ICE services through the "-8" mechanism and is presented in the browser input section. A cookie keyword will be used as the variable name and the normal naming conventions for MAPPER variable naming will apply.

Example #15 and example #16 in the examples category will provided examples of how to set and re-set cookies.

### 2.3.6 Uploading Files

This describes the implementation of requirement #28 specified in section 1.2 plus additional functionality derived from the requirements.

Upload capability for uploading files from the client workstation via HTML Forms via the browser has been implemented.

When the HTML Form attribute ENCTYPE="multipart/form-data" is specified on the HTML Form tag and the TYPE="file" is specified on the Form INPUT tag, the browser will allow the user to browse files on the workstation and attach a file to the form data being send up to the web server.

The following is an example of the HTML Form specifications:-

```
[FORM ENCTYPE="multipart/form-data"
  ACTION="<Gurl>/<category>/service" METHOD=POST]
Filename On Your Computer: [INPUT NAME="WsFile" TYPE="file"]
[INPUT TYPE="submit" VALUE="Send File"]
[/FORM]
```

In the above example the TYPE="file" is causing TWO parameters to be passed into the Cool ICE service via the service input report (-8):-

- <Wsfile> containing the name of the file on the workstation.
- <FUWsFile> containing the name of the file on the server. This is the name of the File after it has been Uploaded to the server.

As the example indicate, Cool ICE will add the prefix "FU" (File Upload) to the name of the input field

name (WsFile).

An uploaded file is only stored temporarily on the server. The Cool ICE Gateway will remove the file when the receiving Cool ICE service returns a response/document to the browser.

The receiving Cool ICE service is expected to copy or move the file to a permanent location as appropriate for the functionality of the service.

Example #18 in the examples category provide an example of how to specify the HTML Form and how to receive and store the file in a Cool ICE service.

### 2.3.7 Downloading Files

This describes the implementation of requirement #28 specified in section 1.2 plus additional functionality derived from the requirements.

Downloading a text file from the server to the workstation can be achieved from a Cool ICE Service. The Cool ICE service must specify the content type of the file being passed down to the browser. The Content Type can be specified in the header of the text file as in the following example:

```
line 1, column 1: Content-Type: application/my-app
line 2            data
etc.              more data
```

Example #19 in the example category will provide example of how a Cool ICE service creates a file for download.

## 2.4 Deployment

### 2.4.1 Enhanced Session Management

This describes the implementation of requirement #28 specified in section 1.2 plus additional functionality derived from the requirements.

Session Management will be enhanced. The Cool ICE system including the gateway has been enhanced to provide a more secure way of handling session id's. The SessionId is no longer part of the URL. SessionID's are kept as cookies. The cookies are created without an expiry so they are not written to disk and they only last while the browser is running. Each SessionId is unique and is not related to any other session, past or present. Each SessionId is paired with the TCP/IP address and is verified for every transaction.

For the browser user this mean that he/she can **bookmark** access to preferred services. When a session is established and the user select a bookmark, the system will not ask the user to signon. Only when a session is not established will the user be asked to signon.

### 2.4.2 Controlling System Availability and System Trace

This describes the implementation of requirements #25 and #32 specified in section 1.2 plus additional functionality derived from the requirements.

The Cool ICE Service Handler will be updated to check for the System Setting in table 27E. When the Cool ICE system has been taken off-line a user defined system availability message will be send back to the browser user.

### 2.4.3 Enhanced Graphics Support

This describes the implementation of additional functionality derived from the requirements.

Available Graphic Chart types will be expanded to include the following new types of charts:-
- Pie, Bar, Gant, Line, Log Line, Polar, Area, Scatter, Tape and Buble.
-

List of possible chart types is available in report 6B of the Cool ICE cabinet.

The Cool ICE Graphing Interface Routine located in 9B of the Cool ICE cabinet has been enhanced to handle a call from a remote Cool ICE service.

Creating Cool ICE Dynamic Graphs will be enhanced to allow specifying the size of a graph. The graphics interface routine located in 9B of the Cool ICE cabinet will be given a new entry point with additional parameters:-

```
@002:(<ChTyp>,<Height>,<Width>,<Title>,<GphNbr>,\
    <ImgPath>,<TempDir>,<TransId>,<GphImgFile>,<status>) .
```

Height and Width parameters have been added. Height and Width is specified in twip's. F.ex. a height,width of 2000,2000 equals 89,134 pixels.

Default height,width is 5850,8600 which will be applied when no size is specified.

Minimum height,width is 1000,1000.

Example #15 in the example category delivered with Cool ICE provides guidance for how to specify the size of a graph.

A STATUS parameter has been has been added to the interface routine. This is used to provide feedback to the calling service as to whether the graph was produced successful or not. The status parameter contains the value 0 (zero) when the graph is produced successful. The status parameter contains the value contains 9 when an error occurred while producing the graph and the GphImfFile parameter points to an error image.

### 2.4.4 Positional URL Parameters

This describes the implementation of requirement #26 specified in section 1.2 plus additional functionality derived from the requirements.

Handling of positional parameters specified on the URL will be improved. Parameters specified on the URL without keyword will be interpreted as positional parameters and the Cool ICE Service Handler will generate default keywords as follows:-

**Error! Reference source not found.**

This URL will generate the following input to the service:-

```
@ldv,p <Category>s8='Examples' .
@ldv,p <Service>s5='Form1' .
@ldv,p <Param1>s5='hello' .
@ldv,p <Param2>s5='world' .
```

### 2.4.5 Service Error Analyser via the Browser

This describes the implementation of additional functionality derived from the requirements.

A new Cool ICE System Service "SvcDump" will be included to view an error dump from a service that failed. The Cool ICE Service Handler will be enhanced to display a message with a link to the new "SvcDump" when a service terminates abnormally. This gives the service developer the capability to view the service error dump using a browser.

The link to the "SvcDump" service will only be available if the security profile of the browser user allows access to the "SvcDump" service.

### 2.4.6 Additional System Variables

This describes the implementation of additional functionality derived from the requirements.

Additional Run-Time information about the environment will be made available to Cool ICE Services. The Cool ICE Service Handler will be updated to supply the following additional Cool ICE system variables in the services input (-8) being passed to a service:-

- <GSysCat> Name of Cool ICE System category (f.ex.ICEADM)
- <GSvrName> Name of Web Server (f.ex. coolice.au.unisys.com)
- <GSvrProtocol> Protocol (f.ex. HTTP/1.0)
- <GSvrPort> Server Port number (f.ex. 80)
- <GRemoteAddr> IP Address of client (f.ex. 129.223.41.102)
- <GUserAgent> Client Browser (f.ex. Mozilla/3.01 (WinNT: I)
- <GGatewayIn> Gateway Input File
- <GInvkSignon> Invoke Signon (y/n)

### 2.4.7 Support for Image Maps

This describes the implementation of additional functionality derived from the requirements.

The Cool ICE Service Handler will be updated to support the HTML ISMAP's. The Cool ICE service will receive two variables (<X> and <Y>) containing the coordinate position of the mouse pointer when the user click within an image defined with the ISMAP attribute.

The Cool ICE Service Handler will be enhanced to create two variables (<X> and <Y>) in the service input report containing the corresponding x and y coordinate position of where the user clicked on the map.

### 2.4.8 Service Expiration Date Enforcement

This describes the implementation of requirement #60 specified in section 1.2 plus additional functionality derived from the requirements.

The Cool ICE Service Handler will be enhanced to verify if a service or object being requested from the Cool ICE Repository has expired. This is utilizing the "Expiration Date" attribute.

### 2.4.9 Automatic Object Download

This describes the implementation of requirements #1 and #28 specified in section 1.2 plus additional functionality derived from the requirements.

The Cool ICE Service Handler will be enhanced to support downloading objects to the browser. The download will be handled transparent to service, so no special scripting is necessary. Any object stored in the Cool ICE repository will be downloaded, this can be an object such as PPT's, XLS's, DOC's, etc. A Cool ICE service just provides a link to an object using the normal HTML Anchor Tag.

All examples in the example category will take advantage of this feature. All examples will include a link to the "Powered by Cool ICE" image which will be stored in the Cool ICE repository.

### 2.4.10 Image Viewer

This describes the implementation of requirement #1 specified in section 1.2 plus additional functionality derived from the requirements.

A new Cool ICE System Service "DspImg" will be included to view an image stored in the Cool ICE Repository. The Cool ICE Menu Builder will be enhanced to include a link to the new "DspImg" service when an object is a GIF or JPG type and it is direct accessible. This provides the capability to view images in the Cool ICE Repository just by changing the accessibility attribute to "Direct".

### 2.4.11 User Password Change

This describes the implementation of requirement #2 specified in section 1.2 plus additional functionality derived from the requirements.

A new Cool ICE System Service "Change-Password-Form" will be included to allow Cool ICE browser users to change their own password. A user must be a registered Cool ICE user and be allowed to change password in order to use the "Change-Password" service.

A new Cool ICE System Service "SignOff" will be included to allow Cool ICE users to sign off from Cool ICE. After Sign Off, the Cool ICE Sign-On form will be displayed. This means a user can safely leave the workstation without the workstation having a session open to Cool ICE.

### 2.4.12 Cookies Support

This describes the implementation of requirement #28 specified in section 1.2 plus additional functionality derived from the requirements.

The Cool ICE Service Handler will be updated to support cookies. Cookies can be set by individual Cool ICE services and the cookie keywords and associated values will be passed into the Cool ICE services. Cookies will be passed into the Cool ICE services through the "-8" mechanism and is presented in the browser input section. A cookie keyword will be used as the variable name and the normal naming conventions for MAPPER variable naming will apply.

Example #15 and example #16 in the examples category will provided examples of how to set and re-set cookies.

### 2.4.13 Repository Search Capability

This describes the implementation of requirement #22 specified in section 1.2 plus additional functionality derived from the requirements.

Function for the browser user will be added to allow searching for Cool ICE services available in the Cool ICE Repository. The search engine allow search for words, phrases using wildcard (*) and boolean operators (and, or, not).

The result of a search will be a list of service titles that match the query. Only services that match the user security profile will be listed.

The Search function will be implemented as a Cool ICE service ("SearchForm"). The Search function will be accessible from the browser and available from the main Cool ICE menu. The Search function is only available if the user security profile provide access to the function.

### 2.4.14 Event Viewer

This describes the implementation of requirement #22 specified in section 1.2 plus additional functionality derived from the requirements.

A Cool ICE Event Viewer will be implemented. It will provide a browser interface for viewing the Cool ICE Event Log.

The Event Viewer will be accessible from the main Cool ICE browser menu for users who have been granted access via the user profile security.

The Event Viewer will display a list of Log Report and a number of options for viewing the log information. One or more Log Reports can be selected for viewing.

### 2.4.15 Uploading Files to the Repository

This describes the implementation of requirement #28 specified in section 1.2 plus additional functionality derived from the requirements.

The Cool ICE Service Handler will be updated to provide support for File Upload as described in section 2.3.6.

### 2.4.16 Enhanced Handling of HTML Tag Delimiters

This describes the implementation of additional functionality derived from the requirements.

The Cool ICE Service Handler will be updated to pass anything between [SCRIPT ...] and [/SCRIPT] as it is and NOT translate the HTML Tag delimiters from [] to <>.

The Source Viewer service will be improved to make a better decision whether a line is a Cool ICE Script or an HTML Tag particular when there is no @ character in column one.

## 2.5 *Discontinued Functionality*

### 2.5.1 Static Document Outside Cool ICE

The storage of Static Documents in directories outside the Cool ICE system has been removed. This has been replaced by the new feature (Requirement #1 that provide functionality to upload and store Static Documents in the Cool ICE Repository.

As a result of this, functions for maintaining Document Directory Aliases has also been removed.

### 2.5.2 Transfer Cool ICE System Images

The "Transfer Cool ICE System Images" function within "Image Directory Alias" dialog has been removed. With Cool ICE 1.1 all system images are stored in the Cool ICE Repository. Cool ICE 1.1 will be taking advantage of the new image handling capabilities introduced with this release. System images will be stored in "ICEADM" system category.

# 3. Revision History

This table is for tracking the date and reason for revisions.

| Rev. | Release Date | Inspection Number | Reason or Description |
|---|---|---|---|
| - | Sept. 30, 1996 | See QIT min. for 96/09/19 | Original Release |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

# Appendix A. User Description

## *Defining Users of System & their needs*

## *Buyers*

## *Key Stake holders*

## *Users*

## *User Characteristics*

## *User Segments*

## $+Cool ICE 1.1



The Cool ICE Administration module:

- provides an environment for administrating a company's Internet/Intranet services
- assists with the development of these Internet/Intranet services
- provides capabilities for handling the deployment of Internet/Intranet services

Cool ICE Administration is primarily targeted at service developers who will be developing, maintaining and enhancing Internet/Intranet services. Functions are also provided for non-technical people to publish static documents over the Internet/Intranet.

The Cool ICE system is a Repository-based system that provides a single point of control for a company's Internet/Intranet assets. Cool ICE Administration provides a point and click environment for maintaining services and their categories in the Repository.

Cool ICE Administration is implemented using Unisys's 4GL Mapper software and uses the power of Mapper for development as well as deployment.

The Cool ICE on-line help contains the following main topics:

- Categories
- Objects
- Event Viewer
- Security
- Options

---

$ Cool ICE 1.1
+ HDK:00001
# HDK3B9ACA40

## $K+Categories

#A category is a collection of services and objects. It is a way of grouping related objects which together create an application, or it may be used to group related topics.

The following illustrates the structure of categories in the Cool ICE Repository:

```
Cool ICE System
    Category A
        Service A1
        Service A2
.....Image I1
        ...
    Category B
      Service B1
```

A category can be set up either as a local category or as a remote category.

- A local category means that all the services and objects for the category will physically reside on the same server within the local Cool ICE system.
- A remote category means that all the services and objects for the category will physically reside on a remote server within the Cool ICE system on that remote server, interconnected via the Cool ICE networking capabilities. Although a category is remote, the Repository information is kept in the central Repository on the local system.

Cool ICE Administration will automatically handle the necessary networking to store and retrieve services and objects belonging to remote categories.

Subtopics are:

- New Category
- Delete Category
- Category Properties
- Rename Category
- Export Category
- Import Category

---

$ Categories
K category
+ HDK:00002
# HDK3B9ACA60

## $K+New Category

#The New Category option is used to create a new category. Setting up a new category involves allocating a drawer in the Mapper system database as the areas for storing the services for this category. The drawer for a category on the local Mapper system will be generated automatically.

Setting up a new category requires that the user have Mapper Coordinator privileges which give permissions to allocate and generate a new drawer in the Mapper database.

In order to create a category on a remote Mapper Server system, the drawer for the category must have been allocated and generated by the coordinator of the remote Mapper system. The drawer must be generated as free form and 80 characters wide.

Category Name
A unique name to identify the category within the Repository.

Category Title
A 70 character long title that will be shown in the list of categories viewed from the browser.

Category Drawer
The Mapper cabinet and drawer to be allocated for storing the services within the category. To locate an available Mapper cabinet and drawer click Browse.

Category Options

- Cool ICE System Category: This option is used to nominate a category to be the Cool ICE System Category. Cool ICE requires a category to be available containing specific services related to the deployment of Cool ICE. A category named "ICEADM" containing default system services is delivered with Cool ICE. If an installation want to customize the system services, it is recommended to copy the system category to another category before applying modifications. This option should then be used to nominate this new category as the System Category for this Cool ICE system. Only one system category is allowed for a Cool ICE system. It will therefore be necessary to remove the system specification on the current category before nominating another category.
- Direct accessible from menu: When checked, this identifies that the category should appear in the menu displayed at deployment time in the browser.
- Available on the Net: This option allows for easily taking a whole category Off the Net while doing maintenance to a range of services within the category. By default, a new category is Off the Net until services are provided within the category, at which time the category should be put On the Net using the Category Properties option.
- Read only: When checked, this option will identify the category as a read only category. This mean that services and objects in this category cannot be updated using the Cool ICE Administration functions. This is to prevent accidental updates to a category.

Category Icon
This allows to select an icon from the Cool ICE repository to be shown next to this category in the menu displayed at deployment time in the browser. You click on the browse button to browse and select the icon from the Cool ICE Repository. The Default Icon button will reset the icon to the system default. The default category icon can be configured using the Cool ICE System Settings function.

Enable Remote Category
When checked, indicates that the category is to be located on a remote Mapper system. A list of remote Mapper systems, which have been configured in the Networking Configuration report (1C2), will be displayed for selection. Cool ICE Administration will verify that the cabinet and drawer specified for the Services Drawer has been generated on the remote Mapper system and is empty. The coordinator of the remote Mapper system must have allocated and generated the drawer as a free form 80 character wide drawer.

Enable User Authentication

---

$ New Category
K Availability;Category Drawer;Category Icon;Category Name;Category Title;Enable Remote Services;Enable User Authentication;New Category;Services Drawer;System Category
+ HDK:00003
# HDK3B9ACA70

This is related to remote categories. When checked, it means that in order for an end user's browser to gain access to a service within this category, their user-id must be passed to the remote Mapper system and validated on the remote system. This provides a high level of security, but requires additional administration. When unchecked, a default user-id and password can be specified and this will then be used for all access to the services within the category. This user-id and password must be defined on the remote system.

## $K+Delete Category

#The Delete Category option is used for deleting a category. Categories located on the local Mapper system will be deleted from the Repository. The Mapper drawer containing the services will also be physically deleted from the system. For remote categories, only the Repository information will be deleted.

---

$ Delete Category
K Delete Category
+ HDK:00004
# HDK3B9ACA80

## $K+Category Properties

#The Category Properties function allows you to select a category and view or change the properties.

---

$ Category Properties
K Category Properties
+ HDK:00005
# HDK3B9ACA90

## $K+View Category Properties

#The View Category Properties dialog allows you to change the Name, Title, Options and Icon of a category.

For description of individual fields, please see New Category function.

Category Drawer and Remote Server Location
These cannot be changed. In order to move a category to another location, you must create a new category and use the Copy Service option to copy the services to the new category.

---

$ View Category Properties
K Availability;Category Drawer;Category Icon;Remote Server Location;Rename;Services Drawer;View Category Properties
+ HDK:00006
# HDK3B9ACAA0

## [$K+]Rename Category

[#]The Rename Category option allows the administrator to change the Repository name of a category. Cool ICE Administration will automatically change the name everywhere it appears in the Repository. For local categories the drawer name will also change where used by the Mapper system.

---

[$] Rename Category
[K] Rename Category
[+] HDK:00007
[#] HDK3B9ACAB0

## $K+Export Category

#The Export Category option provides an easy way of saving the services and objects of a whole category to another media. This can be used for exchanging a category with another Cool ICE system, or simply for backing up the service and object specifications for a category.

Category to Export
The category you want to export.

Directory to Export to
The directory where the category and associated service will be written to. This can be a directory on the hard drive of your local workstation, or simply a diskette on your workstation for easy exchange with another Cool ICE system. A directory can only contain one exported category. If attempting to export to a directory already containing a category, a warning will be given and you will receive the option to overwrite.

Cool ICE Administration will export all the Repository information for the category and the associated services and objects and will export the service specifications themselves including any associated Style Guide information.

---

$ Export Category
K Category to Export;Directory to Export to;Export Category
+ HDK:00008
# HDK3B9ACAC0

## $K+Import Category

#The Import Category option will import a category that previously has been exported from another Cool ICE system.

Category to Import into
This must be an empty category. Cool ICE Administration will import all services and objects included in the category. If you only want to import selected services, we recommend that you import the category to a temporary category and then use the Copy Service option to copy the selected services from there. If the category to import into already contains objects, you will receive a warning and the option to overwrite. The import function will use the original database report number and replace objects in the same database report number.

Directory to Import from
This is the directory where the category was previously exported to. This can be a directory on the hard drive of your local workstation, or simply the diskette drive on your workstation.

---

$ Import Category
K Category to Import into;Directory to Import from;Import Category
+ HDK:00009
# HDK3B9ACAD0

## $K+Objects

#Object is a common term for those services and objects you are keeping in the Cool ICE Repository and is providing on the Internet/Intranet.

An object can be one of the following:

- Documents with static content such as company information and product descriptions
- Inquiry forms and fill in forms such as order entry forms
- Dynamic services providing access to databases and building HTML documents dynamically
- Service Templates for assisting the service development containing commonly used script methods.
- Images and icons
- Applets

Subtopics are:

- Object Attributes
- New Repository Object
- Open Repository Object
- Close Repository Object
- Save Repository Object
- SaveAs Repository Object
- Copy Repository Object
- Delete Repository Object
- Rename Repository Object

---

$ Objects
K Objects
+ HDK:00010
# HDK3B9ACAE0

## $K+Object Attributes

#The attributes of an object are stored in the Cool ICE Repository together with its actual object/service specifications. You can use the Object Attributes dialog to modify these attributes. This dialog will be displayed when opening an existing service or starting a new service.

You specify the name of a new object when you save it to the Repository.

Object
This is the repository name of the object. It is specified when you save the object to the repository.

Category
This is the category to which the object belong within the repository. The category is specified when you save the object to the repository.

Location
An indicator for the physical database location of the object. This is determined by the system when you save the object to the repository.

Object Type
Indicator that show the type of the object. The object type is determined when creating a new object and cannot be changed.

Object Title
A 70 character long title that will be shown in the list of objects/services viewed from the browser.

Availability
An option for easily taking an object/service Off the Net while updating the object. By default, a new service is Off the Net until service specifications have been updated, at which time the service should be put On the Net.

Expiration Date
Specifies when the object becomes unavailable to the browser users. On or after the date specified, browser users who attempt to request the object will receive a message indicating the object is no longer available.

Access to Object
Identifies how the browser user can access the object. Menu means that this object is a service which is responsible for displaying the menu of available services to the browser. When Menu is not chosen, the system will build and display a default menu of available objects within a category. Direct means a link to the object will be provided in the default menu built and displayed by the system. In-Direct means the object can only be requested indirectly from other services/objects.

Object Icon
This allows to select an icon from the Cool ICE repository to be shown next to this object in the menu displayed at deployment time in the browser. You click on the browse button to browse and select the icon from the Cool ICE Repository. The Default Icon button will reset the icon to the system default. The default object icon can be configured using the Cool ICE System Settings function.

Object Specifications
The content of this frame will vary depending on the type of object:

Service Specifications.
This frame will be displayed when the object type is Dynamic. By clicking on the document icon you will get access to the service specifications. The service specifications are Cool ICE Scripting functions. The service is displayed as a Cool ICE Database result report and can be edited using Cool ICE Database editing functions. Normal rules for editing a Cool ICE Database result report apply which for example means that the content of the current displayed report at the time you return will be returned to Cool ICE Administration and will be

---

$ Object Attributes
K Access to Object;Availability;Expiration Date;Export HTML;Import HTML;Object Attributes;Object Icon;Object Title;Object Type;Service Specs;Trace On
* HDK:00011
# HDK3B9ACAF0

regarded as updates to the service. Using functions such as Find or Locate should be avoided as they are native Cool ICE Database functions and causes that you cannot return to Cool ICE Administration.

Trace On
Provides a powerful debugging aid for tracing problems in a dynamic service. When turned on, the service request from the browser, including all input to the service, will be saved in a trace log.

Export HTML
If the template contain HTML, this is used to export the HTML to be maintenance by an HTML Authoring tool.

Import HTML
Used for importing HTML from an HTML Authoring tool to the service.

Document Specifications
This frame will be displayed when the object type is Static Document. By clicking on the Modify Document icon you will get access to the document specifications. The document is displayed as a Cool ICE Database result report and can be edited using Cool ICE Database editing functions. Normal rules for editing a Cool ICE Database result report apply which for example means that the content of the current displayed report at the time you return will be returned to Cool ICE Administration and will be regarded as updates to the service. Using functions such as Find or Locate should be avoided as they are native Cool ICE Database functions and causes that you cannot return to Cool ICE Administration.

Export Document
This is used to export the Document to be maintenance by an HTML Authoring tool.

Import Document
Used for importing a Document from an HTML Authoring tool to the document.

Image Specifications
This frame will be displayed when the object type is Image.

Export Image
This will export the image to a file on your workstation. You can the open and edit the image by an Image Editor tool.

Import Image
Used for importing an image from a workstation file and replace the current image.

Template Specifications.
This frame will be displayed when the object type is Template. By clicking on the Modify Template icon you will get access to the template specifications. The template specifications are Cool ICE Scripting functions. The template is displayed as a Cool ICE Database result report and can be edited using Cool ICE Database editing functions. Normal rules for editing a Cool ICE Database result report apply which for example means that the content of the current displayed report at the time you return will be returned to Cool ICE Administration and will be regarded as updates to the service. Using functions such as Find or Locate should be avoided as they are native Cool ICE Database functions and causes that you cannot return to Cool ICE Administration.

Export HTML
If the template contain HTML, this is used to export the HTML to be maintenance by an HTML Authoring tool.

Import HTML
Used for importing HTML from an HTML Authoring tool to the template.

Applet Specifications
This frame will be displayed when the object type is Applet.

Export Applet
Exporting an applet is not available.

Import Applet
Used to import an updated version of the applet from a workstation file.

## [$K+]New Repository Object

[#]The following is a list of different objects you can insert in the Cool ICE Repository:

- Dynamic Service
- Image
- Applet
- Template
- Static Document

---

[$] New Repository Object
[K] New Repository Object
[+] HDK:00017
[#] HDK3B9ACEA0

## $K+Dynamic Service

#The Dynamic Service option starts the creation of a new dynamic service. The following options for new dynamic services are available:

Dynamic Service based on Cool ICE Script Template
Opens a dialog box providing access to browse the Cool ICE repository to select an appropriate template which you want to use as the basis for the new dynamic service.

Dynamic Service based on HTML Template
Allows the service developer to combine the use of HTML authoring tools with the Cool ICE Scripting development environment. This will import templates previously created with an HTML authoring tool into the Cool ICE service run to form the basis for developing a dynamic service. Inquiry Forms, Input Forms and Output Tables are examples of HTML templates able to be developed using an authoring tool.

Dynamic Service based on existing script
Allows to browse the Cool ICE database to select an existing script report which you want to use as the basis for developing a new dynamic Cool ICE service.

---

$ Dynamic Service
K Dynamic Service based on HTML Template;Dynamic Service based on ICE Script Template;Dynamic Service
+ HDK:00018
# HDK3B9ACB00

## [$K+]Select Template

[#]The Select Template function allows you to browse the Cool ICE Repository selecting a Cool ICE Template object. Only objects previously inserted into repository as Cool ICE Templates using the new template function will be listed.

Use default Cool ICE Template
When checked, this option will select the Cool ICE default template. This template contains the very basic format for a Cool ICE dynamic service which is the format all Cool ICE services must conform to.

---

[$] Select Template
[K] Select Template
[+] HDK:00019
[#] HDK3B9ACE90

## $K+Import HTML Template

#The Import HTML Template option is associated with the Dynamic Service based on HTML Template option, which you can select when creating a new dynamic service. The following describes what happens during the import process:

- HTML tags are mapped to the Cool ICE Style Guide. The HTML template is scanned and selected HTML tags and attributes are translated into the equivalent Style Guide keywords. The mapping between the HTML tags and Style Guide keywords is described in table 30E. Maintenance of this table is only provided through the use of manual Mapper editing options.
- Images are uploaded to the Web Server. The HTML template is scanned for references to images which are then transferred to either the Cool ICE Repository or the image directory on the web server.
- Long lines are automatically wrapped. Lines longer than 80 characters are wrapped at a space or at the end of an HTML tag. When that is not possible, a line will be broken at column 80 and continued on the next line.
- Essential Mapper run statements are inserted. The necessary Mapper run statements to turn the HTML template into a dynamic service and to interface to the Cool ICE system are inserted at the top and bottom of the template.
- HTML tag delimiters are converted. HTML tag delimiters (less than and greater than) are converted to opening and closing square brackets. This is to avoid confusion with the standard Mapper variable name delimiters (less than and greater than). These delimiters will automatically be converted back by the Cool ICE Service Handler before a document is sent to the browser.
- Anything between <SCRIPT...> and </SCRIPT> such as JavaScript or VBScript which is not Cool ICE script will be passed untouched. JavaScript/VBScript typically contain special characters such as <> and [] and special characters like these will not be converted.
- The Action attribute on the HTML FORM tag will be modified to the Cool ICE standard for referencing another service from an html form. <GUrl>/<category>/COOL-ICE-SERVICE will be inserted in the action field. This means the service developer just have to modify and replace COOL-ICE-SERVICE with the appropriate Cool ICE service name.

Workstation HTML File
Specifies where on your local workstation the HTML template is to be imported from.

Browse
Allows you to browse through directories to find the HTML template.

Map HTML Tags to Cool ICE Style Guide
Allows you to turn the mapping to the Style Guide on and off.

Upload Images
Allows you to turn uploading of images referenced within the HTML template on and off. Based on the reference to the image as specified in the HTML tag <IMG>, Cool ICE will attempt to determine the location of the image. If the image resides on a drive not accessible from your workstation, you will be asked to specify the exact location or skip the upload. If the image file already exists on the server you will be asked to confirm replacing the image. Multiple references to the same image are not detected and you will be asked to confirm replacing the image.

The following option are available for uploading images:

- Save in Cool ICE Repository. This means that images being uploaded to the server will be inserted in the Cool ICE repository. This option is recommended to use when you want to manage images by the Cool ICE Administration environment for things like securing, backing up and exporting/importing images. When selecting this option you will be allowed to specify in which Cool ICE Repository Category to save the uploaded image.

---

$ Import HTML Template
K Browse;Import HTML Template;Map HTML Tags to ICE Style Guide;Server Image Directory;Upload Images to Web Server;Workstation HTML File;import
+ HDK:00020
# HDK3R0ACR10

- Save in Server Directory. This mean that images being uploaded to the server will be inserted in an image directory on the server outside of the Cool ICE database. This option is recommended to use when performance of sending images to the browser is more important than managing the images by the Cool ICE Administration environment. When selecting this option you will be allowed to select the directory to upload the image to. A list of valid server image directories is displayed for selection.

## $K+Import Existing Script Report

#The Import Existing Script Report option allows you to browse the Cool ICE database to select an existing database report to be used as the basic for a new dynamic Cool ICE service.

Script Report Location
This allows you to specify the location of an existing script report. If you know the report identification you can specify it directly. The Browse button allows you to browse the Cool ICE database and select an existing report.

Apply default Cool ICE script envelope
When checked the standard Cool ICE script envelope will be applied to the script report. The script envelope is Cool ICE script statements necessary for a dynamic service to interface to the Cool ICE run time environment. Script will be inserted at the top and bottom of the script report.

---

$ Import Existing Script Report
K Import Existing Script
+ HDK:00023
# HDK3B9ACF80

## $K+Upload Image File

#The Upload Image File dialog is displayed when the upload process of an HTML template cannot determine the location of an image to be transferred from the workstation to the Web server.

You are asked to identify the location of the image or skip upload of this image.

---

$ Upload Image File
K Upload Image File
+ HDK:00021
# HDK3B9ACB20

## $K+Save Uploaded Object

#As a result of uploading images to the Cool ICE repository this dialog box will appear asking you to select in which category you want to insert the image and you are asked to provide the Object Name.

Cancel
Selecting the cancel button will skip the upload of the image.

---

$ Save Uploaded Object
K Save Uploaded Object
+ HDK:00022
# HDK3B9ACEC0

## [$K+]Import Image

[#]The Import Image function is used when inserting a new image in the Cool ICE Repository and when importing new revision to replace an existing image in the Cool ICE Repository.

Workstation Image File
Specifies where on your local workstation the image is to be imported from.

Browse
Allows you to browse through directories to find the image file.

Image Type
Allow you to specify the type of image if it is different from the file extension.

---

[$] Import Image
[K] Import Image
[+] HDK:00024
[#] HDK3B9ACED0

## [$K+]Import Applet

[#]The Import Applet function is used when inserting a new applet in the Cool ICE Repository and when importing new revision to replace an existing applet in the Cool ICE Repository.

Workstation Applet File
Specifies where on your local workstation the applet is to be imported from.

Browse
Allows you to browse through directories to find the applet file.

---

[$] Import Applet
[K] Import Applet
[+] HDK:00025
[#] HDK3B9ACE00

## [$K+]Insert Template

[#]The Inserting Template function is used when inserting a new template in the Cool ICE Repository. You are allowed to browse through the Cool ICE Repository to select an existing service or template to be used as the basic for the new template.

---

$ Insert Template
K Insert Template
+ HDK:00026
# HDK3B9ACEE0

## [$K+]Import Static Document

[#]The Import Static Document option is associated with the New Repository Object option. The following describes what happens during the import process: .

- Images are uploaded to the Web Server. The HTML document is scanned for references to images which are then transferred to either the Cool ICE Repository or the image directory on the web server.
- Long lines are automatically wrapped. Lines longer than 80 characters are wrapped at a space or at the end of an HTML tag. When that is not possible, a line will be broken at column 80 and continued on the next line.
- Because this is a static document, no Cool ICE Script functions are inserted.
- HTML tag delimiters are not converted.

Workstation HTML File
Specifies where on your local workstation the HTML document is to be imported from.

Browse
Allows you to browse through directories to find the HTML document.

Upload Images
Allows you to turn uploading of images referenced within the HTML document on and off. Based on the reference to the image as specified in the HTML tag <IMG>, Cool ICE will attempt to determine the location of the image. If the image resides on a drive not accessible from your workstation, you will be asked to specify the exact location or skip the upload. If the image file already exists on the server you will be asked to confirm replacing the image. Multiple references to the same image are not detected and you will be asked to confirm replacing the image.

The following option are available for uploading images:

- Save in Cool ICE Repository. This means that images being uploaded to the server will be inserted in the Cool ICE repository. This option is recommended to use when you want to manage images by the Cool ICE Administration environment for things like securing, backing up and exporting/importing images. When selecting this option you will be allowed to specify in which Cool ICE Repository Category to save the uploaded image.
- Save in Server Directory. This mean that images being uploaded to the server will be inserted in an image directory on the server outside of the Cool ICE database. This option is recommended to use when performance of sending images to the browser is more important than managing the images by the Cool ICE Administration environment. When selecting this option you will be allowed to select the directory to upload the image to. A list of valid server image directories is displayed for selection.

---

[$] Import Static Document
[K] Import Static Document
[+] HDK:00027
[#] HDK3B9ACEE0

## $K+Export HTML

#The Export HTML option is used when you want to use an HTML authoring tool for maintaining and updating the HTML template within a service. The HTML section of the service will be exported to a file on your workstation. This file can then be opened in the authoring tool and updated using features of that tool. When finished the template should be saved to the file ready for being imported back into the service.

Everything within the service between the start and end HTML tag ([HTML] and [/HTML]) will be exported. Any Mapper scripting options will be included and identified using the HTML script tag [SCRIPT] and associated end tag [/SCRIPT]. The position of script options relative to HTML tags should be maintained when updating the HTML template using the authoring tool. If required, the script options can be updated using the authoring tool as all of the HTML template including script options will be imported back into the service using the Import HTML option.

Any script language such as JavaScript or VBScript identified between [SCRIPT...] and [/SCRIPT] will be passed untouched. JavaScript/VBScript typically contain special characters such as <> and [] and special characters like these will not be converted.

In case the Dynamic Service you are exporting from contain more than one HTML template, a dialog box will appear requesting you to identify and select which of the HTML templates you want to export.

References to the Style Guide within the HTML template will be replaced with their respective values from the Style Guide for the current service.

Workstation HTML File
Specifies where on your local workstation the HTML template is to be exported to.

Browse
Allows you to browse through directories to find a file name.

---

$ Export HTML
K Export HTML
+ HDK:00012
# HDK3B9ACB30

## $K+Import HTML

#The Import HTML option is used for importing an HTML template into the current open service after having exported it for editing in an authoring tool. The HTML template being imported will replace everything within the current open service between the start and end HTML tag ([HTML] and [/HTML]). The following describes what happens during the import process:

- HTML tags are mapped to the Cool ICE Style Guide. The HTML template will be scanned and selected HTML tags and attributes translated into the equivalent Style Guide keywords. The mapping between the HTML tags and Style Guide keywords is described in table 30E. Maintenance of this table is through the use of manual Mapper editing options.
- Images are uploaded to the Web Server. The HTML template is scanned for references to images which are then transferred to either the Cool ICE Repository or the image directory on the web server.
- Long lines are automatically wrapped. Lines longer than 80 characters are wrapped at a space or at the end of an HTML tag. When that is not possible, a line will be broken at column 80 and continued on the next line.
- HTML tag delimiters are converted. HTML Tag delimiters (less than and greater than) will be converted to opening and closing square brackets. This is to avoid confusion with the standard Mapper variable name delimiters (less than and greater than). These delimiters will automatically be converted back by the Cool ICE Service Handler before a document is sent to the browser.
- Anything between <SCRIPT...> and </SCRIPT> such as JavaScript or VBScript which is not Cool ICE script will be passed untouched. JavaScript/VBScript typically contain special characters such as <> and [] and special characters like these will not be converted.
- The Action attribute on the HTML FORM tag will be modified to the Cool ICE standard for referencing another service from an html form. <GUrl>/<category>/COOL-ICE-SERVICE will be inserted in the action field. This means the service developer just have to modify and replace COOL-ICE-SERVICE with the appropriate Cool ICE service name.

In case the Dynamic Service you are importing to contain more than one HTML template, a dialog box will appear requesting you to identify and select which of the HTML templates you want to replace.

Workstation HTML File
Specifies where on your local workstation the HTML template is to be imported from.

Browse
Allows you to browse through directories to find the HTML template.

Map HTML Tags to Cool ICE Style Guide
Allows you to turn the mapping to the Style Guide on and off.

Upload Images to Web Server
Allows you to turn uploading of images referenced within the HTML template on and off. Based on the reference to the image as specified in the HTML tag <IMG>, Cool ICE will attempt to determine the location of the image. If the image resides on a drive not accessible from your workstation, you will be asked to specify the exact location or skip the upload. If the image file already exists on the server you will be asked to confirm replacing the image. Multiple references to the same image are not detected and you will be asked to confirm replacing the image.

Server Image Directory
Identifies the directory on the Internet/Intranet server to which the image should be transferred. A list of valid server image directories is displayed.

---

$ Import HTML
K Browse;Import HTML;Map HTML Tags to ICE Style Guide;Server Image Directory;Upload Images to Web Server;Workstation HTML File
+ HDK:00013
# HDK3B9ACB40

## $K+Select HTML Template

#The Select HTML Template dialog box appears when a dynamic service contains more than one HTML template. When exporting HTML from a dynamic service or importing HTML to a dynamic service containing more than one HTML template, you will be asked to identify which HTML Template you want.

---

$ Select HTML Template
K Select HTML Template
+ HDK:00014
# HDK3B9ACE10

## [$K+]Export Document

[#]The Export Document option is used when you want to use an HTML authoring tool for maintaining and updating the static HTML document. The HTML document will be exported to a file on your workstation. This file can then be opened in the authoring tool and updated using features of that tool. When finished the document should be saved to the file ready for being imported back into the Cool ICE Repository.

Workstation Document File
Specifies where on your local workstation the HTML Document is to be exported to.

Browse
Allows you to browse through directories to find a file name.

---

[$] Export Document
[K] Export Document
[+] HDK:00015
[#] HDK3B9ACF50

## [$K+]Export Image

[#]The Export Image option is used when you want to use an Image Editing tool for editing the image. The Image will be exported to a file on your workstation. This file can then be opened in the Image Editor and updated using features of that tool. When finished the image should be saved to the file ready for being imported back into the Cool ICE Repository.

Workstation Image File
Specifies where on your local workstation the Image is to be exported to.

Browse
Allows you to browse through directories to find a file name.

---

[$] Export Image
[K] Export Image
[+] HDK:00016
[#] HDK3B9ACF60

## [$K+]Open Repository Object

[#]The Open Object option allows you to browse the Cool ICE Repository for selecting an object to open. The object will be issued and locked to your workstation. Any other workstation attempting to open the same object will get a message indicating that the object is already issued to your workstation. Objects located within remote categories will automatically be pulled across from the remote system and presented on your workstation. Also, remote objects will be locked to the workstation opening the object.

Read Only
Allows the developer to open an object that is locked to another workstation in read only mode. If an object opened in read only mode is modified, it can only be saved in the Repository under a new name.

---

$ Open Repository Object
K Open Repository Object;Read Only
+ HDK:00028
# HDK3B9ACB50

## [$K+]Close Repository Object

[#]The Close Object option releases the object which is currently open and locked to your workstation. It will verify if the object attributes including object specifications have changed and if necessary save the object back into the Repository and close the Object Attributes dialog.

---

[$] Close Repository Object
[K] Close Repository Object
[+] HDK:00029
[#] HDK3B9ACB60

## [$K+]Save Repository Object

[#]The Save Object option saves changes to object attributes including object specifications back into the Repository. The object remains locked to your workstation.

---

[$] Save Repository Object
[K] Save Repository Object
[+] HDK:00030
[#] HDK3B9ACB70

## $K+SaveAs Repository Object

#The SaveAs Object option allows you to save an object under a new name and also specify in which category the object will be saved.

---

$ SaveAs Repository Object
K SaveAs Repository Object
+ HDK:00031
# HDK3B9ACB80

## $K+Copy Repository Object

#The Copy Object option allows you to copy one or more objects from one category to another or within the same category.

The object attributes and object specifications, including associated Style Guide objects, will be copied.

Cool ICE Administration provides copying to and from remote categories and will automatically handle all the necessary networking between remote servers. Copying directly between two remote categories is not supported. In order to copy from one remote category to another remote category we recommended that you create a category on the local system to be used for temporary intermediate storage.

Copy From
Allows you to browse through categories. Within a category, you may select one or more services to copy by highlighting the objects in the list.

Select All
Used for selecting all objects in a category. The objects in the list will not be highlighted but they will all be selected for copying and will appear in the Confirm Copy Object dialog.

Copy To
Allows you to select the category to which you want to copy the objects.

---

$ Copy Repository Object
K Copy From;Copy Repository Object;Copy To;Select All
+ HDK:00032
# HDK3B9ACB90

## $K+Confirm Copy Object

#The Confirm Copy Object dialog allows you to confirm copying of all selected objects or of individual objects from the list.

Yes
Confirms copying the highlighted object only.

Yes to All
Confirms copying all the objects in the list.

No
Removes an object from the list.

---

$ Confirm Copy Object
K Confirm Copy Object
+ HDK:00033
# HDK3B9ACBA0

## $K+Confirm Replace Object

#The Confirm Replace Object dialog is displayed in case an object name already exists in the category being copied to.

Replace
Allows you to confirm Replace. The existing object will be overwritten.

Replace All
Allows you to confirm Replace for all objects. All existing objects with duplicated names will be overwritten.

No
Skips copying the object.

New Name
Allows you to specify a new name for an object with a duplicated name.

---

$ Confirm Replace Object
K Confirm Replace Object;New Name
+ HDK:00034
# HDK3B9ACBB0

## [$K+]Delete Repository Object

[#]The Delete Object option allows you to delete one or more objects from a category.

Browse through available categories and select a category. Within a category, select one or more objects to delete by highlighting the objects in the list.

Select All
Used for selecting all objects in a category. The objects in the list will not be highlighted but they will all be selected for deletion and will appear in the Confirm Delete Object dialog.

---

$ Delete Repository Object
K Delete Repository Object
+ HDK:00035
# HDK3B9ACBC0

## $K+Confirm Delete Object

#The Confirm Delete Object option allows you to confirm deletion of all selected objects or remove individual objects from the list.

The object attributes and object specifications, including associated Style Guide objects, will be deleted.

Yes
Confirms deleting the highlighted object only.

Yes to All
Confirms deleting all the objects in the list.

No
Removes an object from the list.

---

$ Confirm Delete Object
K Confirm Delete Object
+ HDK:00036
# HDK3B9ACBD0

## $K+Rename Repository Object

#The Rename Object option allows you to change the Repository name of an object.

Browse through available categories and select a category. Within a category, select the object to rename by highlighting the object in the list.

---

$ Rename Repository Object
K Rename Repository Object
+ HDK:00037
# HDK3R9ACBE0

## [$K+]Confirm Rename Object

[#]The Confirm Rename option allows you to specify a new name for the object. The name of the object will be changed wherever it is used within the Cool ICE Repository.

---

[$] Confirm Rename Object
[K] Confirm Rename Object
[+] HDK:00038
[#] HDK3B9ACBE0

## [$K+]Event Viewer

[#]The Event Viewer option allows you to view events logged by Cool ICE. The following types of event logs are kept:

- Access Log. Logs all requests for services made from a browser.
- Error Log. Logs any service failure due to a syntactical error in the service specifications.
- Trace Log. Records information for services that have the Trace On attribute checked.

---

[$] Event Viewer
[K] Event Viewer
[+] HDK:00039
[#] HDK3B9ACC00

## $K+Access Log

#The access log stores details of service requests made from a browser and logged by Cool ICE (who requested the service, when was the service requested, and how long did it take for Cool ICE to execute the service). Some uses for this information are as follows:

- It can provide details of the services that are of most interest to users, which can be used for marketing purposes.
- It can be used to charge for Internet services.
- It can provide details about when the server is busy.

The Access Log option has the following suboptions for specifying both the time period you wish to analyze and how you want to view the information:

- Access Log Settings
- View Access Log

---

$ Access Log
K Access Log
+ HDK:00040
# HDK3B9ACC10

## $K+Access Log Settings

#The access log is maintained on a daily basis. A new log report is allocated each day, and will contain all service requests logged for that day. At the first service request of a new day, Cool ICE will summarize the log data of the previous day and add the data to a summary report.

This summary report is kept at report 8F within the drawer where Cool ICE is installed. Access to the information in the summary report is not provided by any specific Cool ICE Administration option, but it can be accessed through Mapper for individual customized analysis.

The log cycle can be configured as follows:

- Daily Cycle. The Log report will be overwritten every day.
- Weekly Cycle. Log reports will be kept on a weekly basis.
- Monthly Cycle. Log reports will be kept on a monthly basis.
- Logging Off. Logging is turned off and service requests will not be recorded.

---

$ Access Log Settings
K Access Log Settings
+ HDK:00041
# HDK3B9ACC20

## [$K+]View Access Log

[#]The View Access Log option allows you to select the day you want to analyze and how you want to view the information.

The following standard views are provided:

- Most Popular Services
- Average Service Times
- User Sessions
- Services by User
- Users by Service
- Peak Time Hits

---

[$] View Access Log
[K] View Access Log
[+] HDK:00042
[#] HDK3B0ACC30

## $K+Most Popular Services

#This option provides a summary of the number of hits (service requests) per category and service. The Cool ICE Sign-On service is excluded from the list.

---

$ Most Popular Services
K Most Popular Services
+ HDK:00043

## $K+Average Service Times

#This option provides the average service time per category and service. This is the elapsed time it took to perform the service. It is measured from the time at which the Cool ICE Service Handler received the request to the time at which it sent the HTML document back to the Web Server. The Cool ICE Sign-On service is excluded from the list.

---

$ Average Service Times
K Average Service Times
+ HDK:00044
# HDK3B9ACC50

## $K+User Sessions

#This option provides a summary of Cool ICE sessions initiated by the users.

---

$ User Sessions
K User Sessions
+ HDK:00045

## $K+Services by User

#This option provides a summary of the number of hits (service requests) per category and service listed by user-ids. The Cool ICE Sign-On service is excluded from the list.

---

$ Services by User
K Services by User
+ HDK:00046

## $K+Users by Service

#This option provides a summary of the number of hits (service requests) per user-id listed by category and service. The Cool ICE Sign-On service is excluded from the list.

---

$ Users by Service
K Users by Service
+ HDK:00047
# HDK3R9ACC80

## $K+Peak Time Hits

#This provides a list of the total number of hits (service requests) for the selected time interval. The default time interval is 60 minutes. You can also select 30 and 15 minute intervals.

---

$ Peak Time Hits
K Peak Time Hits
+ HDK:00048
# HDK3B9ACC90

## [$K+]Error Log

[#]Whenever a service fails because of a syntactical error in the service specifications, the error will be logged in an error log by Cool ICE. The error log provides the service developer with a lot of information about the service at the time it failed (the date and time of the failure, who requested the service and internal values from the service).

The View Error Log option displays entries in the error log.

---

[$] Error Log
[K] Error Log
[+] HDK:00049
[#] HDK3B9ACCA0

## $K+View Error Log

#The View Error Log option displays the entries recorded in the error log. Selecting a log entry from the list will show the information logged for the incident.

View
Shows the log information for the entry highlighted in the list.

Delete
Deletes the log entry highlighted in the list.

Clear All
Deletes all log entries in the list.

Log entries are not deleted automatically by Cool ICE. After the service developer has investigated the incident they must delete it.

---

$ View Error Log
K View Error log
+ HDK:00050
# HDK3B9ACCB0

## $K+Trace Log

#The Trace Log option is a powerful debugging aid for the service developer who needs to trace and debug a dynamic service in order to analyze a problem. Services requested through the browser are executed in background mode and only output in HTML format can be viewed by the browser. The service developer therefore normally has no means of tracing and debugging a service in the browser environment.

The trace log enables the service developer to run a service in foreground using the standard Mapper debugging facilities, such as the Run Debugger, as well as the simple Display options.

The trace log is associated with the Trace On feature within the Service Attributes dialog. When turned on, Cool ICE will log all the input for a service each time it is requested by the browser. When the service request and all input is captured, the service can be run in the foreground, simulating the input coming from a browser.

The View Trace Log option displays entries in the trace log.

---

$ Trace Log
K Trace Log
+ HDK:00051

## $K+View Trace Log

#The View Trace Log option displays entries in the trace log. Selecting a log entry from the list will show the information logged for the incident.

View
Shows the log information for the entry highlighted in the list.

Run It
Opens a dialog for running normal Mapper foreground runs, and runs the service as a foreground run. Prior to this, the service developer should have updated the service specifications to include appropriate debugging options such as RDB and DSP.

Delete
Deletes the log entry highlighted in the list.

Clear All
Deletes all log entries in the list.

Log entries are not deleted automatically by Cool ICE. After the service developer has investigated the incident they must delete it.

---

$ View Trace Log
K Run It;View Trace Log
+ HDK:00052
# HDK3B9ACCD0

## $K+Security

#Cool ICE provides security at run time when services are requested from browsers. At run time, the security profile of the user is matched with the security profile of the service being requested. If the profiles match, the user will be allowed to perform the service.

The basic concept is that only certain users need to be registered in the system — this approach reduces the administrator burden.

The security mechanism has been designed for ease of maintenance. By default, the system is open and services can be requested by everyone with access to Cool ICE. Only for those services where restrictions apply will it be necessary to specify a security profile — a matching profile must be allocated to those users, who will be granted permission.

Security profiles can be specified at the system, category or service level. Profiles specified at a certain level will automatically be inherited by services below that level. Security profiles at the system level will apply to all services in the whole system. Security profiles specified for a category will apply to all services within that category. Security profiles specified for a service will only apply to that service.

Procedure for setting up security

1. Create security profiles
2. Register users and assign profiles to users through profile membership
3. Allocate profiles to individual Categories or Services

The following is a list of security maintenance options:

- Security Profiles
- User Registration
- Services Access Security

---

$ Security
K Security
+ HDK:00053
# HDK3B9ACCE0

## $K+Security Profiles

#The Security Profiles option is used for setting up a table of valid profiles to be allocated to users and services. A profile must be specified in the Security Profile table before it can be allocated.

Add
Opens a dialog for adding a new profile to the list.

Modify
Opens a dialog for modifying the description for the highlighted profile.

Remove
Deletes the highlighted profile from the list. A profile can only be deleted if it is not allocated to any users or services.

Allocate Profile
Allows a profile to be allocated to a range of users.

Where Used
Displays a report of how the profiles have been allocated.

---

$ Security Profiles
K Security Profiles
+ HDK:00054
# HDK3B9ACCF0

## [$K+]Add-Modify Security Profile

[#]The Add-Modify Security Profile option dialog allows you to add or modify a security profile.

Profile
The name of the profile to add or modify.

Description
A short description of the profile.

---

[$] Add-Modify Security Profile
[K] Add-Modify Security Profile
[+] HDK:00055
[#] HDK3B9ACD00

## $K+Security Profiles Where Used

#The Security Profiles Where Used option provides an overview of the profiles and how they have been allocated to users and categories/services.

Profile Allocation
Opens the Profile Allocation dialog, allowing you to allocate the highlighted profile to a range of users. This dialog is also displayed by double clicking on a profile in the list.

---

$ Security Profiles Where Used
K Security Profiles Where Used
+ HDK:00056
# HDK3B9ACD10

## $K+Security Profile Allocation

#The Security Profile Allocation option allows you to allocate or de-allocate a profile to a range of users.

Add
Takes users highlighted in the Not Allocated to list and adds them to the Allocated to list.

Remove
Removes users highlighted in the Allocated to list and puts them back in the Not Allocated to list.

---

$ Security Profile Allocation
K Security Profile Allocation
+ HDK:00057

## $K+User Registration

#The User Registration option is used for registering users for whom special security profiles need to be allocated.

Not all users need to be registered to use Cool ICE. Users that are not registered in Cool ICE will be allowed access to all open services. A service is open, if it does not have special security profiles allocated.

Users who have been registered and have been give membership of selected profiles, will be given access to all open services and services with a matching profile.

Add
Opens a dialog for adding a new user to the list.

Modify
Opens a dialog box allowing you to modify the user registration of the user highlighted in the list.

Remove
Deletes the highlighted user id from the list.

Profile Membership
Allows you to give a user membership of a range of profiles.

Profile Report
Displays a report of the users and the profiles they are members of and the categories/services they have been granted access to.

---

$ User Registration
K Profile Membership;Profile Report;User Registration
+ HDK:00058
# HDK3B9ACD30

## $K+Add-Modify User Registration

#The Add-Modify User Registration option dialog is used for adding new Cool ICE users to the list of authorized users.

User Id
The user-id to add. This can be selected from the list of users registered in the Mapper environment. A Cool ICE user must be a registered user of the Mapper system. If a user is not yet a registered user in the Mapper environment, Cool ICE Administration will automatically perform the registration with a default set of permissions; enough to allow the user to use Cool ICE.

Department
The department number in which this user is registered within the Mapper environment. A department is a way of grouping users with common interests.

User cannot change password
Checking this option means the user will not be able to change their password. It is strongly recommended that you check this option for general public userid's (e.g., Guest). When this option is unchecked, the browser user will be able to change their password by selecting the "Change Password" function from the main Cool ICE browser window.

---

$ Add-Modify User Registration
K Add-Modify User Registration;Department;User Id
+ HDK:00059
# HDK3B9ACD40

## $K+User Security Profile Report

#The User Security Profile Report option dialog provides an overview of the users and the profiles they are member of and which categories/services they have been granted special access to.

Profile Allocation
Opens the Profile Membership dialog, allowing you to give the highlighted user membership to a range of profiles. This dialog is also displayed by double clicking on a user in the list.

---

$ User Security Profile Report
K Profile Allocation;User Security Profile Report
+ HDK:00060
# HDK3B9ACD50

## $K+User Security Profile Membership

#The User Security Profile Membership option dialog allows you to give a user membership of a range of profiles or to take profile memberships away from a user.

Add
Adds profiles highlighted in the Not Member of list to the Member of list.

Remove
Remove profiles highlighted in the Member of list and puts them back in the Not Member of list.

---

$ User Security Profile Membership
K User Security Profile Membership
+ HDK:00061

## $K+Object Access Security

#The Object Access Security option is used for allocating security profiles for selected objects for which special security is required.

Not all objects need to be allocated profiles in order to be accessible by users. All users will be given access to all open objects. An object is open if it does not have special security profiles allocated.

Objects which have been allocated profiles will only be accessible by users with a matching profile.

Inheritance Hierarchy
This is used for selecting the level at which you wish to specify security. Security can be specified at different levels: a specific object, a category, or the whole system. Security specified at the system level will automatically apply to all objects in the whole system. Security specified for a selected category will only apply to objects within that category. Security specified for a single object will only apply to that object.

Add
Opens a dialog box for adding a profile to the list.

Modify
Opens a dialog box for modifying the highlighted profile entry.

Remove
Deletes the highlighted profile entry.

Re-Inherit
Resets a profile with the profile defined for the level above it in the hierarchy.

Profile Report
Displays a report of the categories and objects showing the security profiles which have been allocated and the user who have been granted access.

---

$ Object Access Security
K Inheritance Hierarchy;Object Access Security
+ HDK:00062
# HDK3B9ACD70

## $K+Add-Modify Object Access Security

#The Add-Modify Object Access Security dialog is used for adding or modifying profiles for authorized access to an object.

The Profile 1, Profile 2 and Profile 3 columns allow you to select the profiles to be allocated. This is a list of valid profiles created using the Security Profile option. You can specify how the profiles are to be matched against the user profiles. You can also specify logical and/or conditions, as follows:

- Selecting and inserting 'RealEstate' from Profile 1 and also selecting and inserting 'Solicitor' from the same Profile 1, ensures that this object can be accessed only by users who are 'RealEstate' or by users who are 'Solicitor'.
- Selecting and inserting 'RealEstate' from Profile 1 and 'Solicitor' from Profile 2, this object can be accessed only by users who are 'RealEstate' and 'Solicitor'.

---

$ Add-Modify Object Access Security
K Add-Modify Object Access Security
+ HDK:00063
# HDK3B9ACD80

## $K+Object Security Profile Report

#The Object Security Profile Report option dialog provides an overview of the categories and objects and the profiles which have been allocated and which users have been granted special access.

---

$ Object Security Profile Report
K Object Security Profile Report
+ HDK:00064

## $K+Options

#The Options menu provides the following options for configuring Cool ICE:

- Style Guide
- Arrange Categories/Objects
- Image Directory Aliases
- Browser SignOn Configuration
- Cool ICE System Settings
- Graphics Server Settings

---

$ Options
K Options
+ HDK:00065

## $K+Style Guide

#The Style Guide option serves two roles, as follows:

- To allow you to provide a consistent look and feel to HTML documents generated by a dynamic service, and to be able to change this at one source.
- To provide an easy way of maintaining common variables to be used in dynamically generated HTML code. The use of a Style Guide provides an easy way for maintaining variables commonly used by all services in the whole system or just a range of services. If a variable needs to be changed, you only need to change it in the Style Guide and it will automatically have effect at run time in all services referencing the variable.

The Inheritance Hierarchy list shows the hierarchy of categories and services within Cool ICE. At the top of the category is Cool ICE. You can specify a Style Guide for the whole system, for a category or for a service. A Style Guide specified at the system level will automatically apply to all services in the system. A Style Guide specified for a selected category will only apply to a service within that category. A Style Guide specified for a single service will only apply to that service.

Add
Opens a dialog box for adding a keyword to the list.

Modify
Opens a dialog box for modifying the highlighted keyword in the list.

Remove
Deletes the highlighted keyword in the list.

Re-Inherit
Resets the Style Guide with the Style Guide defined for the level above it in the hierarchy.

---

$ Style Guide
K Style Guide
+ HDK:00066
# HDK3B9ACDA0

## $K+Add-Modify Style Guide Keyword

#The Add-Modify Style Guide Keyword option dialog is used to add or modify keywords in the list of Style Guide keywords.

Keyword
A unique entry of an item in the Style Guide. At run time the Cool ICE Service Handler will create a Mapper variable out of the keyword containing the value specified in the Value field. A service can then use the keyword where appropriate just be referencing the equivalent variable.

Value
The content of the item in the Style Guide.

Description
A short description of the entry.

The Color Chart button displays a color chart.

---

$ Add-Modify Style Guide Keyword
K Add-Modify Style Guide Keyword;Description;Keyword;Value
+ HDK:00067

## $K+Color Chart

#The Color Chart is used for selecting one of the 16 basic colors for a Style Guide entry. The color value is returned as a red, green, blue (RGB) hexadecimal value.

---

$ Color Chart
K Color Chart
+ HDK:00068
# HDK3B9ACDC0

## $K+Arrange Categories/Objects

#The Arrange Categories and Objects option is used for manipulating the order in which you want Categories and Objects to be listed when displayed for the browser user and also when browsing the Cool ICE Repository using the Cool ICE Administration functions.

Arrange Objects
This will open a dialog box displaying a list of objects within the highlighted category allowing you to arrange objects within that category.

Move
This will identify and select the highlighted category as the one to be moved. The category to be moved will be shown in the field Category to move.

Insert Before
This will move and insert the category shown in the field Category to move before the category highlighted in the list of categories.

Insert After
This will move and insert the category shown in the field Category to move after the category highlighted in the list of categories.

---

$ Arrange Categories/Objects
K Arrange Categories/Objects
+ HDK:00069
# HDK3B9ACE30

## $K+Arrange Objects

#The Arrange Objects option is accessed from the Arrange Categories/Objects function and is used for manipulating the order in which you want Objects to be listed when displayed for the browser user and also when browsing the Cool ICE Repository using the Cool ICE Administration functions.

Move
This will identify and select the highlighted object as the one to be moved. The object to be moved will be shown in the field Object to move.

Insert Before
This will move and insert the object shown in the field Object to move before the object highlighted in the list of objects.

Insert After
This will move and insert the object shown in the field Object to move after the object highlighted in the list of objects.

---

$ Arrange Objects
K Arrange Objects
+ HDK:00070
# HDK3B9ACF40

## $K+File Transfer

#The File Transfer option is provided primarily for transferring images between the service developers workstation and the Cool ICE web server. However, the file transfer will allow transfer of any file.

Images being transferred to the web server using the file transfer option will be stored in directories outside the Cool ICE Repository. As described for Image Directory Aliases, storing images in directories outside the Cool ICE Repository mean images are not secured and managed by the Cool ICE system. This should only be used when performance of sending images to the browser is more important than managing the images by the Cool ICE Administration environment.

Server
In this frame you select server directory alias and/or files you want to transfer to/from depending on which transfer function you select (Transfer from Server or Transfer from Workstation).

Server Alias Directory
This is a list of Image Directory Aliases as setup using the Image Directory Aliases function. When selecting an alias, a list of files within the corresponding directory will be displayed. The selection of server directories is limited to directories setup using the Image Directory Aliases function.

Select files or Select All
When transferring files from server to workstation you will select the files to transfer either by highlighting individual files from the list or check the Select All box for selecting all files within the list. When using the Select All option the files will not be highlighted.

Workstation
In this frame you select workstation directory and/or files you want to transfer to/from depending on which transfer function you select (Transfer from Server or Transfer from Workstation).

Directories
This is a list of directories on your workstation. When selecting a directory, a list of files within that directory will be displayed.

Select files or Select All
When transferring files from workstation to server you will select the files to transfer either by highlighting individual files from the list or check the Select All box for selecting all files within the list. When using the Select All option the files will not be highlighted.

Transfer from Server
Selecting this button will transfer files from the Cool ICE web server to the service developers workstation. Files to transfer must be selected within the Server frame. A dialog box will be opened to show a list of files selected and ask for a confirmation to start the transfer.

Transfer from Workstation
Selecting this button will transfer files from the service developers workstation to the Cool ICE web server. Files to transfer must be selected within the Workstation frame. A dialog box will be opened to show a list of files selected and ask for a confirmation to start the transfer.

---

$ File Transfer
K File Transfer
+ HDK:00071
# HDK3B9ACE70

## $K+File Transfer from Server to Workstation

#This dialog box allows you to confirm the file transfer. A list of files selected within the specified server alias directory is listed requesting you to confirm the transfer.

Transfer Now
Selecting this button will start the transfer from the Server to the Workstation.

---

$ File Transfer from Server to Workstation
K File Transfer from Server to Workstation;Transfer from Server to Workstation
+ HDK:00072

## $K+File Transfer from Workstation to Server

#This dialog box allows you to confirm the file transfer. A list of files selected within the specified workstation directory is listed requesting you to confirm the transfer.

Transfer Now
Selecting this button will start the transfer from the Workstation to the Server.

---

$ File Transfer from Workstation to Server
K File Transfer from Workstation to Server;Transfer from Workstation to Server
+ HDK:00073

## $K+Image Directory Aliases

#The Image Directory Aliases option allows you to maintain a table of mappings to directories on the Web Server allocated for storing image files. This is used when you are importing HTML template and documents and check the Uploading Images to the Web Server button. This table provides a list of valid image directories into which images can be uploaded.

Storing images in these image directories mean that images are kept outside the Cool ICE Repository and as such they are not secured and managed by the Cool ICE system. This option is recommended to use only when performance of sending images to the browser is more important than managing the images by the Cool ICE Administration environment.

Add
Opens a dialog box for adding a new alias to the list.

Modify
Opens a dialog box for modifying the highlighted alias in the list.

Remove
Deletes the highlighted alias from the list.

---

$ Image Directory Aliases
K Image Directory Aliases;Transfer Cool ICE System Images
+ HDK:00074
# HDK3B9ACDE0

## $K+Add-Modify Image Directory

#The Add-Modify Image Directory option dialog is used to add or modify a server image directory. Cool ICE uses these directories for storing images on the server.

Alias
A unique short name for an image directory. The alias must be defined in the Web Server as a valid alias. On some Web Servers (for example, Netscape Web Server) this is also called a Prefix. The alias is specified in the URL and the Web Server translates it into the physical path.

Server Image Directory
The physical path name of the directory.

Default Cool ICE System Image Directory
This is checked for the directory that Cool ICE will use for storing images referenced specifically by Cool ICE. That is, images and icons used in Cool ICE menus and system error massages. Only one directory can be the default directory.

---

$ Add-Modify Image Directory
K Add-Modify Image Directory;Alias;Default Cool ICE System Image Directory;Server Image Directory
+ HDK:00075
# HDK3B9ACE00

## $K+Browser SignOn Configuration

#The Browser SignOn Configuration option allows you to configure how users coming into Cool ICE from a browser will identify themselves. When the Cool ICE Gateway has been configured to request user sign-on, the actual sign-on form will be requested from the Cool ICE System. This dialog allows you to configure the content of the sign-on form.

Provided you keep the basic content as it exists in the system sign-on form, you can redesign the look and feel of the form using an HTML authoring tool and replacing the system sign-on form.

Request Department Number
Shows an input box in the form, requesting the user to type in the Mapper department number their user-id belongs to.

Hide Department Number
Hides the input box and as such it will not request the user to specify their Mapper department number. In this case all valid Cool ICE users must belong to the same Mapper department. This department number is specified in the department field of the Guest User-Id frame.

User-Id - Department and Password
When specified, will pre-fill the sign-on form to indicate a default user-id which can be used for casual users.

---

$ Browser SignOn Configuration
K Browser SignOn Configuration;Hide Department Number;Request Department Number;User-Id - Department and Password
+ HDK:00076
# HDK3B9ACE10

## $K+Cool ICE System Settings

#The Cool ICE Systems Settings functions access to Cool ICE options at the system level. These are deployment options which have effect at runtime when a user is accessing the Cool ICE system from a browser.

System Availability
This allows to take the whole Cool ICE system off-line and on-line. If Cool ICE is taken off-line, the user accessing Cool ICE from the browser will receive a message indicating the system is not available.

Service Handler Trace
This allows to turn trace on/off for the Cool ICE service Handler. Service Handler trace information is kept in the F-drawer of the cabinet where Cool ICE is installed. Report 2F contains the input received from the Gateway. Report 3F contains output received from the service being called. Report 4F contains the service input report created by the Service Handler. These three reports will be reused and overwritten for every service call.

Category Icon
This allows to setup the preferred default icon to be used when displaying a list of categories to the browser user. You click on the browse button to browse and select the icon from the Cool ICE Repository. The Reset button will reset to the icon before selecting this function.

Object Icon
This allows to setup the preferred default icon to be used when displaying a list of objects to the browser user. You click on the browse button to browse and select the icon from the Cool ICE Repository. The Reset button will reset to the icon before selecting this function.

---

$ Cool ICE System Settings
K Cool ICE System Settings
+ HDK:00077
# HDK3B9ACE40

## [$K+]Graphics Server Settings

[#]Configuring the Graphics Server is only required for Cool ICE systems running on Unixware.

The Graphics Server software on Cool ICE systems on Unixware is running on a PC and a request to produce a graph is passed from Cool ICE to the Graphics Server via a TCP/IP connection. Cool ICE must know on which PC the Graphics Server software is installed and on which socket the Graphics Server will receive a request.

Name or IP address of PC running Graphics Server
This specifies either the name or the IP address of the PC where the Graphics Server software is running.

Socket number configured in the Graphics Server
This specifies the socket number configured in the Graphics Server. The default number is 1234.

---

$ Graphics Server Settings
K Graphics Server Settings
+ HDK:00078
# HDK3B9ACE50

## [$K+]Miscellaneous

[#]This following topics document miscellaneous dialog boxes used in Cool ICE Administration.

- Server Directory Inquiry
- Workstation Browser
- Cool ICE Database Drawer Browser
- Cool ICE Database Report Browser
- Cool ICE Repository Browser
- Cool ICE System Variables

---

$ Miscellaneous
K Miscellaneous
+ HDK:00079
# HDK3B9ACE70

## $K+Server Directory Inquiry

#Cool ICE will attempt to verify the existence of directories on the Cool ICE server by passing a directory inquiry on to the Operating System. In order for the Operating System to allow such a request a userid and password is required which allow access to the OS. On NT systems, the userid being used for starting the Mapper NT software must have the user right "Log on as a batch job" in order to pass a DOS command to NT.

On a system configured with the above mentioned user right, only users who are connecting in to the Cool ICE Administration environment via a remote connection such as MSW (Mapper System for Windows) will be required to specify a userid/password. The userid/password entered will be remembered by the system and as long as the userid stays valid you will not be asked to provide this again for any subsequent server directory inquiry. The userid is kept on a COOLICE.INI file on your workstation. When the userid becomes invalid, you will be asked to specify a new valid userid/password.

Server UserId
This is a UserId on the Operating system (NT UserID or Unix UserID).

On NT systems this userid must have the user right to "Log on as a batch job" and be a member of the Mapper Group.

Server Password
This is a password associated with the userid.

---

$ Server Directory Inquiry
K Server Directory Inquiry
+ HDK:00080
# HDK3B9ACE30

## $K+Workstation Browser

#This allows you to browse and select directories and files on your workstation or network drives your workstation have access to.

You select a directory by double click on a directory in the list of directories.

You select a file by double click on a file in the list of files.

Preview
The preview button will display the content of a file you previously have selected by double click on a file in the list of files. This can only display content of files with .TXT, .HTM or .HTML extension.

---

$ Workstation Browser
K Workstation Browser
+ HDK:00081
# HDK3B9ACE60

## $K+Cool ICE Database Report Browser

#This Cool ICE Database Report Browser is displayed when selecting to browse the Cool ICE database. It allows you to browse and select an existing report from the database.

You select a Cool ICE Database Drawer by double click on a drawer in the list of drawers.

You select a Cool ICE Database Report by double click on a report in the list of reports.

---

$ Cool ICE Database Report Browser
K Cool ICE Database Report Browser
+ HDK:00082
# HDK3B9ACEB0

## $K+Cool ICE Repository Browser

#This allows you to browse and select an object from the Cool ICE Repository.

You select a Cool ICE Category by double click on a category in the list of categories.

You select a Cool ICE Object by double click on an object in the list of objects.

---

$ Cool ICE Repository Browser
K Cool ICE Repository Browser
+ HDK:00083
# HDK3B9ACF20

**$K+Cool ICE System Variables**

#

{bmc t:_ourwo~1\joshua\online~1\00000002.wmf}

---

$ Cool ICE System Variables
K Cool ICE System Variables
+ HDK:00084
# HDK3B9ACFB0

## [$K+]Cool ICE Database Drawer Browser

[#]The Drawer Browser is displayed when selecting browse. It allows you to browse and select an open (available) Cool ICE Database drawer and cabinet to store your Cool ICE category.

An available drawer for storing your Cool ICE category is listed as being OPEN in the Drawer Browser.

You select a drawer by double clicking on an open one in the list of drawers.

---

[$] Cool ICE Database Drawer Browser
[K] Cool ICE Database Drawer Browser
[+] HDK:00085
[#] HDK3B9ACFC0